Final Report for AOARD Grant 114111

"Learnable Models for Information Diffusion and its Associated User Behavior in Micro-blogosphere"

30/08/2012

Name of Principal Investigators (PI and Co-PIs): Kazumi Saito

- e-mail address : k-saito@u-shizuoka-ken.ac.jp
- Institution : School of Administration and Informatics, University of Shizuoka
- Mailing Address : 52-1 Yada, Suruga-ku, Shizuoka 422-8526 Japan
- Phone : +81-54-264-5436
- Fax : +81-54-264-5436

Period of Performance: 08/31/2011 – 08/30/2012

Abstract: Short summary of most important research results that explain why the work was done, what was accomplished, and how it pushed scientific frontiers or advanced the field. This summary will be used for archival purposes and will be added to a searchable DoD database.

First, we addressed the problem of detecting the period in which information diffusion burst occurs from a single observed diffusion sequence under the assumption that the delay of the information propagation over a social network follows the exponential distribution. To be more precise, we formulated the problem of detecting the change points and finding the values of the time delay parameter in the exponential distribution as an optimization problem of maximizing the likelihood of generating the observed diffusion sequence. We devised an efficient iterative search algorithm for the change point detection whose time complexity is almost linear to the number of data points. We tested the algorithm against the real Twitter data of the 2011 Tohoku earthquake and tsunami, and experimentally confirmed that the algorithm is much more efficient than the exhaustive naive search and is much more accurate than the simple greedy search.

Second, we addressed the problem of how people make their own decisions based on their neighbors' opinions. The model best suited to discuss this problem is the voter model and several variants of this model have been proposed and used extensively. However, all of these models assume that people use their neighbors' latest opinions. Thus, we enhanced the original voter model and defined the temporal decay voter (TDV) model incorporating a temporary decay function with parameters, and proposed an efficient method of learning these parameters from the observed opinion diffusion data. We further proposed an efficient method of selecting the most appropriate decay function from among the candidate functions each with the optimized parameter values. We adopted three functions as the typical candidates: the exponential decay, the power-law decay, and no decay, and evaluate the proposed method (parameter learning and model selection) through extensive experiments. We, first, experimentally demonstrated, by using synthetic data, the effectiveness of the proposed method, and then we analyzed the real opinion diffusion data from a Japanese word-of-mouth communication site for cosmetics using three decay functions above, and showed that most opinions conform to the TDV model of the power-law decay function.

Introduction: Include a summary of specific aims of the research and describe the importance and ultimate goal of the work.

We focus on on-line societies including sites such as for micro-blogging, social networking, knowledge-sharing and media-sharing in the World Wide Web, through which behaviors, ideas and opinions can spread over time. Clearly, the information diffusion and its contents evolution processes in these on-line societies also reflects complex social structures and distributed social interests. Thus, it is worth putting some effort to attempt to find empirical regularities and develop explanatory accounts of human communication in these sites. Such attempts would be valuable for understanding social structures and trends, and inspire us to discover new knowledge and provide insights into underlying human communication. Our ultimate goal of this project is to develop learnable models for

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **28 FEB 2013**	2. REPORT TYPE **Final**	3. DATES COVERED **31-08-2011 to 30-08-2012**

4. TITLE AND SUBTITLE
Learnable Models for Information Diffusion and its Associated User Behavior in Micro-blogosphere

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Kazumi Saito

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
School of Administration and Informatics, University of Shizuoka,52-1 Yada,Suruga-ku,Shizuoka 422-8526,NA,NA

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AOARD, UNIT 45002, APO, AP, 96338-5002

10. SPONSOR/MONITOR’S ACRONYM(S)
AOARD

11. SPONSOR/MONITOR’S REPORT NUMBER(S)
AOARD-114111

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

There are two major contributions: 1) a new efficeint and effective method to detect a burst in information diffusion from the observed data, and 2) a new opinion formation model that takes people’s past opininon into account. Both problems are formulated as a maximum liklihood problem in which the likelihood of observing the data from the model is maximized. For 1) the algorithm was tested against the real Twitter data of the 2011 Tohoku earthquake and tsunami, and for 2) the algorithm was tested against the real opinion diffusion data from a Japanese word-of-mouth communication site for cosmetics. Both confirmed that the algorithms are efficient and work as expected.

15. SUBJECT TERMS
Information diffusion, Opinion formation, Social network, Burst detection, Influential nodes, Network dynamics, Knowledge discovery from network

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**140**	

rumor spreading and its associated user behavior in a micro-blogosphere. We believe that our research outcome helps understanding fundamental mechanisms of information diffusion and evolution processes in our society. Moreover, it is highly expected that this kind of mathematical studies using large-scale networks such as a micro-blog communication network can bridge a gap between empirical social networks analyses and fundamental mathematics.

Experiment: Description of the experiment(s)/theory and equipment or analyses.

The information diffusion data we used for evaluation were extracted from 201,297,161 tweets of 1,088,040 Twitter users who tweeted at least 200 times during the three weeks from March 5 to 24, 2011 that includes March 11, the day of 2011 Tohoku earthquake and tsunami. It is conceivable to use a retweet sequence in which a user sends out other user's tweet without any modification. But there exists multiple styles of retweeting (official retweet and unofficial retweet), and it is very difficult to accurately extract a sequence of tweets in an automatic manner considering all of these different styles. Therefore, in our experiments, noting that each retweet includes the ID of the user who sent out the original tweet in the form of "@ID", we extracted tweets that include @ID format of each user ID and constructed a sequence data for each user. More precisely, we used information diffusion sequences of 798 users for which the length of sequences are more than 5,000 (number of tweets). Note that each diffusion sequence includes retweet sequences on multiple topics. Since we do not know the ground truth of the change points for each sequence if there are changes in it, we used the naive method which exhaustively searches for all the possible combinations of the change points as giving the ground truth. We had to limit the number of change points to 2 (J = 2) in order for the naive method to return the solution in a reasonable amount of computation time.

The opinion formation data we used for evaluation were collected from "@cosme", which is a Japanese word-of-mouth communication website for cosmetics. In @cosme, a user can post a review and give a score of each brand (one from 1 to 7). When one user registers another user as his/her favorite user, a "fan-link" is created between them. We traced up to ten steps in the fan-links from a randomly chosen user in December 2009, and collected a set of (b, k, t, v)'s, where (b, k, t, v) means that user v scored brand b k points at time t. The number of brands was 7,139, the number of users was 45,024, and the number of reviews posted was 331,084. For each brand b, we regarded the point k scored by a user v as the opinion k of v, and constructed the opinion diffusion sequence D_{T0} (b) consisting of 3-tuple (k, t, v). In particular, we focused on these brands in which the number of samples N = |D_{T0} (b)| was greater than 500. Then, the number of brands was 120. We refer to this dataset as the @cosme dataset.

Results and Discussion: Describe significant experimental and/or theoretical research advances or findings and their significance to the field and what work may be performed in the future as a follow on project. Fellow researchers will be interested to know what impact this research has on your particular field of science.

Information diffusion: By analyzing the real information diffusion data, we revealed that even if the data contains tweets talking about plural topics, the detected burst period tends to contain tweets on a specific topic intensively. In addition, we experimentally confirmed that assuming the information diffusion path to be the line shape tree results in much better approximation of the maximum likelihood estimator than assuming it to be the star shape tree. This is a good heuristic to accurately estimate the change points when the actual diffusion path is not known to us. These results indicate that it is possible to detect and identify both the burst period and the topic diffused without extracting the tweet sequence for each topic and identifying the diffusion paths for each sequence, and the proposed method can be a useful tool to analyze a huge amount of information diffusion data. Our immediate future work is to compare the proposed method with existing burst detection methods that are designed for data stream. We also plan to devise a method of finding nodes that caused the bust based on the change points detected.

Opinion formation: We first tested the proposed algorithms by synthetic datasets assuming that there are two decay models: the exponential decay and the power-law decay. We confirmed that the learning algorithm correctly identifies the parameter values and the model selection algorithm correctly identifies which model the data came from. We then applied the method to the real opinion diffusion data taken from a Japanese word-of-mouth communication site for cosmetics, i.e., the @cosme dataset. We used the two decay functions above and added no decay function as a baseline. The result of the analysis revealed that opinions of most of the brands conform to the TDV model of the power-law decay function. We found this interesting because this is consistent with the observation that many human actions are related to the power-law. Some brands showed behaviors characteristic to the brands, e.g., the older brand that releases new product less frequently naturally follows no decay TDV and the newer brand that releases new product more frequently naturally follows the power-law decay TDV with large decay constant, which are all well interpretable.

List of Publications and Significant Collaborations that resulted from your AOARD supported project: In standard format showing authors, title, journal, issue, pages, and date, for each category list the following:

a) papers published in peer-reviewed journals,

1. Masahiro Kimura, Kazumi Saito, Kouzou Ohara and Hiroshi Motoda, "Learning to predict opinion share and detect anti-majority opinionists in social networks" to appear in Journal of Intelligent Information Systems (JIIS).
2. Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda, "Detecting Changes in Information Diffusion Pattern over Social Network," to appear in ACM Transactions on Intelligent Systems and Technology (TIST).

b) papers published in peer-reviewed conference proceedings,

1. Shoko Kato, Akihiro Koide, Takayasu Fushimi, Kazumi Saito and Hiroshi Motoda, "Network Analysis of Three Twitter Functions: Favorite, Follow and Mention," to appear in Proc. of the 2012 Pacific Rim Knowledge Acquisition Workshop (PKAW2012).
2. Takayasu Fushimi, Kazumi Saito and Kazuhiro Kazama "Extracting Communities in Networks based on Functional Properties of Nodes," to appear in Proc. of the 2012 Pacific Rim Knowledge Acquisition Workshop (PKAW2012).
3. Masahiro Kimura, Kazumi Saito, Kouzou Ohara and Hiroshi Motoda, "Opinion Formation by Voter Model with Temporal Decay Dynamics," Proc. of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD2012), pp. 565--580, 2012.
4. Kazumi Saito, Kouzou Ohara, Masahiro Kimura and Hiroshi Motoda, "Burst Detection in a Sequence of Tweets based on Information Diffusion Model," Proc. of the Fifteenth International Conference on Discovery Science (DS2012), pp. 239--253, 2012.
5. Kazumi Saito, Masahiro Kimura, Kouzou Ohara and Hiroshi Motoda, "Graph Embedding on Spheres and its Application to Visualization of Information Diffusion Data," Proc. of the International Workshop on Mining Social Network Dynamics (MSND2012), pp. 1137--1144, 2012.
6. Kouzou Ohara, Kazumi Saito, Masahiro Kimura, and Hiroshi Motoda, "Effect of In/Out-Degree Correlation on Influence Degree of Two Contrasting Information Diffusion Models," Proc. of the 2012 International Conference on Social Computing, Behavioral Modeling, and Prediction (SBP2012), pp. 131--138, 2012.
7. Takayasu Fushimi, Yamato, Kubota, Kazumi Saito, Masahiro Kimura, Hiroshi Motoda, and Kouzou Ohara, "Speeding up Bipartite Graph Visualization Method," Proc. of the 24th Australasian Joint Conference on Artificial Intelligence (AI2011), pp.697--706, 2011.

c) papers published in non-peer-reviewed journals and conference proceedings,
d) conference presentations without papers,
e) manuscripts submitted but not yet published, and
f) provide a list any interactions with industry or with Air Force Research Laboratory scientists or significant collaborations that resulted from this work.

None.

Attachments: Publications a), b) and c) listed above if possible.

DD882: As a separate document, please complete and sign the inventions disclosure form.

Important Note: If the work has been adequately described in refereed publications, submit an abstract as described above and cite important findings to your above List of Publications. If a full report needs to be written, then submission of a final report that is very similar to a full length journal article will be sufficient in most cases. This document may be as long or as short as needed to give a fair account of the work performed during the period of performance. There will be variations depending on the scope of the work. As such, there is no length or formatting constraints for the final report. Keep in mind the amount of funding you received relative to the amount of effort you put into the report. For example, do not submit a $300k report for $50k worth of funding; likewise, do not submit a $50k report for $300k worth of funding. Include as many charts and figures as required to explain the work.

J Intell Inf Syst manuscript No.
(will be inserted by the editor)

Learning to predict opinion share and detect anti-majority opinionists in social networks

Masahiro Kimura · Kazumi Saito · Kouzou Ohara · Hiroshi Motoda

Received: date / Accepted: date

Abstract We address the problem of detecting anti-majority opinionists using the value-weighted mixture voter (VwMV) model. This problem is motivated by the fact that 1) each opinion has its own value and an opinion with a higher value propagates more easily/rapidly and 2) there are always people who have a tendency to disagree with any opinion expressed by the majority. We extend the basic voter model to include these two factors with the value of each opinion and the anti-majoritarian tendency of each node as new parameters, and learn these parameters from a sequence of observed opinion data over a social network. We experimentally show that it is possible to learn the opinion values correctly using a short observed opinion propagation data and to predict the opinion share in the near future correctly even in the presence of anti-majoritarians, and also show that it is possible to learn the anti-majoritarian tendency of each node if longer observation data is available. Indeed, the learned model can predict the future opinion share much more accurately than a simple polynomial extrapolation can do. Ignoring these two factors substantially degrade

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-11-4111, and JSPS Grant-in-Aid for Scientific Research (C) (No. 23500194).

Masahiro Kimura
Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
E-mail: kimura@rins.ryukoku.ac.jp

Kazumi Saito
School of Administration and Informatics, University of Shizuoka
Shizuoka 422-8526, Japan
E-mail: k-saito@u-shizuoka-ken.ac.jp

Kouzou Ohara
Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
E-mail: ohara@it.aoyama.ac.jp

Hiroshi Motoda
Institute of Scientific and Industrial Research, Osaka University
Osaka 567-0047, Japan
E-mail: motoda@ar.sanken.osaka-u.ac.jp

the performance of share prediction. We also show theoretically that, in a situation where the local opinion share can be approximated by the average opinion share, 1) when there are no anti-majoritarians, the opinion with the highest value eventually takes over, but 2) when there are a certain fraction of anti-majoritarians, it is not necessarily the case that the opinion with the highest value prevails and wins, and further, 3) in both cases, when the opinion values are uniform, the opinion share prediction problem becomes ill-defined and any opinion can win. The simulation results support that this holds for typical real world social networks. These theoretical results help understand the long term behavior of opinion propagation.

Keywords Social networks · Opinion dynamics · Parameter learning

1 Introduction

The emergence of large scale social computing applications has made massive social network data available, and large networks formed by these services play an important role as a medium for spreading diverse information including news, ideas, opinions, and rumors (Newman et al, 2002; Newman, 2003; Gruhl et al, 2004; Domingos, 2005). Thus, investigating the spread of influence in social networks has been the focus of attention (Leskovec et al, 2007a; Crandall et al, 2008; Wu and Huberman, 2008; Romero et al, 2011; Bakshy et al, 2011; Mathioudakis et al, 2011).

The most well studied problem would be the *influence maximization problem*, that is, the problem of finding a limited number of influential nodes that are effective for spreading information through the network. Many new algorithms that can effectively find approximate solutions have been proposed both for estimating the expected influence and for finding good candidate nodes under different model assumptions, *e.g.*, descriptive probabilistic interaction models (Domingos and Richardson, 2001; Richardson and Domingos, 2002), and basic diffusion models such as the *independent cascade (IC) model* and the *linear threshold (LT) model* (Kempe et al, 2003; Kimura et al, 2010a; Leskovec et al, 2007b; Chen et al, 2009, 2010). This problem has good applications in sociology and “viral marketing” (Agarwal and Liu, 2008). However, the models used above allow a node in the network to take only one of the two states, *i.e.*, either active or inactive, because the focus is on *influence*.

Applications such as an on-line competitive service in which a user can choose one from multiple choices and decisions, however, require a different approach where a model must handle multiple states. The model best suited for this kind of analysis would be a voter model, which is the model to analyze how different opinions spread over a social network. It is one of the most basic stochastic process model, and has the same key property with the *linear threshold (LT) model* in that a node decision is influenced by its neighbor’s decision, *i.e.*, a person changes its opinion by the opinions of its neighbors. In the basic voter model which is defined on an undirected network, each node initially holds one of the two opinions, *e.g.*, yes or no, and adopts the opinion of a randomly chosen neighbor at each subsequent discrete time-step.

In this paper, we address the problem of opinion formation by using an extended voter model for which multiple states are needed. There are three extensions. As described above, the original voter model can handle only two opinions and assumes discrete time-step. We extended the basic voter model to be able to handle K opinions and to allow asynchronous opinion update. This is just to make the basic voter model to be more realistic and this extension is straightforward. Indeed, the actual opinion update is asynchronous and if we are to use the observed data, synchronous discrete time-step model does not work. The other two extensions are more fundamental. We note that when we have to make a decision from multiple choices, we consider the value of each choice, *e.g.*, quality, brand, authority, etc. because this definitely affects our choice. The same is true for opinion formation. We listen to and evaluate what our neighbors say and change our opinions. Thus, the second extension is to incorporate the *value* of each opinion as a new parameter. The extended model is referred to as the *value-weighted voter (VwV) model with multiple opinions*. Same as the basic voter model, the VwV model assumes that people naturally tend to follow their neighbors' majority opinion. However, we note that there are always people who do not agree with the majority and support the minority opinion, which was also addressed in Gill and Gainous (2002) and Arenson (1996). We are interested in how this affects the opinion share. Thus, the third extension is to include this anti-majority effect by linearly combining the VwV model and the anti-majority model with the *anti-majoritarian tendency* of each node as a new parameter. The extended model is referred to as the *value-weighted mixture voter (VwMV) model*. We will discuss how to learn these parameters from the observed opinion propagation data and how accurately the learned model can predict the future opinion share.

There has been a variety of work on the voter model. Liggett (1999) and Sood and Redner (2005) extensively studied dynamical properties of the basic model, including how the degree distribution and the network size affect the mean time to reach consensus, from a mathematical point of view. Castellano et al (2009) and Yang et al (2009) investigated several variants of the voter model and analyzed non-equilibrium phase transition from a physics point of view. Holme and Newman (2006) and Crandall et al (2008) extended the voter model to combine it with a network evolution model. These studies gave insights into the fundamentals of the vote model, but their focuses are different from what this paper intends to address, *i.e.*, parameter learning from the data and share prediction at a specific time T with opinion values and anti-majoritarian tendency considered. Even-Dar and Shapira (2007), whose work we think has a similar goal to ours in spirit, investigated the influence maximization problem (maximizing the spread of the opinion that supports a new technology) at a given target time T under the basic *voter model*, *i.e.*, with two opinions (one in favor of the new technology and the other against it). They showed that the most natural heuristic solution, which picks the nodes in the network with the highest degree, is indeed the optimal solution, under the condition that all nodes have the same cost. This work is close to ours in that it measures the influence at a specific time T, but is different in all others (no share prediction, no value and anti-majoritarian tendency considered, no more than two opinions, no asynchronous update and no learning). We should mention that we are not the first to introduce the notion of anti-majority. There is a model called anti-voter model where only two opinions are considered (Huber and Reinert, 2004; Donnelly and Welsh, 1984; Matloff, 1977). Each one chooses one of its neighbors randomly and decides to take the opposite opinion of the neighbor chosen. Röllin (2007) analyzed the statistical property of the anti-voter model introducing the notion of exchangeable pair couplings. Our work is different from theirs, apart from the learning mechanism and being able to handle multiple opinions, in that we consider the effect of both the voter and the anti-voter models by introducing the *anti-majoritarian tendency* of each node as a new parameter.

This paper is an extension and integration of what we have reported in Kimura et al (2010b) and Kimura et al (2011). In the former we addressed the problem of predicting the opinion share at a future time (before an consensus is reached) by learning the opinion values from a limited amount of past observed opinion diffusion data using the VwV model. In the latter we introduced the VwMV model and mainly focused on the learning performance of the *anti-majoritarian tendency* of each node. In this paper we extend our preliminary work and analyze the share prediction performance of the VwMV model when both the opinion values and the anti-majoritarian tendency are not known and have to be learned from the observed opinion propagation data, investigate how the average anti-majoritarian tendency affects the learning performance, and detect who the anti-majoritarians are. In particular, we seek for the answer for the following questions: what the opinion share will be in the near future, given only the limited amount of observed data, how easy it is to learn both opinion value and anti-majority tendency, and how much the observed data is required to learn and identify the anti-majoritarians accurately enough. It is important to learn the model quickly and predict what will happen in the near future when a new opinion appears. The model is too simple to accurately predict the far future. For this, it is more desirable to understand the asymptotic behavior by a theoretical analysis.

We conjecture that learning opinion values is easy because the number of opinion K is not many (order of tens), but learning anti-majoritarian tendency is not easy because the tendency is associated with each node and the number of nodes is huge (order of ten thousands or more). We further conjecture that predicting the opinion share is much easier than identifying the anti-majoritarians because the former is a macroscopic quantity over the whole network but the latter is defined for each node. We show that both the parameters, anti-majoritarian tendency and opinion value, can be learned by an iterative algorithm that maximizes the likelihood of the model's generating the observed data, and confirmed the above conjectures by experiments. We tested the algorithm for four real world social networks with size ranging over 4,000 to 12,000 nodes and 40,000 to 250,000 links, and showed that the parameter value update algorithm correctly identifies both the values of opinions and the anti-majoritarian tendency of each node under various situations. The opinion values can be learned in good accuracy with a small amount of data, but the anti-majoritarian tendency needs a sufficiently large amount of data to improve the accuracy. Use of the learned model can predict the opinion share in the near future very accurately despite the existence of anti-majoritarians. The theoretical analysis under the assumption in which the local opinion share can be approximated by the average opinion share shows that 1) when there are no anti-majoritarians, the opinion with the highest value eventually takes over, but 2) when there is a certain fraction of anti-majoritarians, it is not necessarily the case that the opinion with the highest value prevails and wins, and further, 3) in both cases, when the opinion values are uniform, the opinion share prediction problem becomes ill-defined and any opinion can win, and these are also supported by real world networks in which the above assumption does not hold. We want to emphasize that it is crucially important to explicitly model the anti-majority effect to obtain good results. Predicting the share by VwV model when there are anti-majoritarians does not work. There seems to be no simple way to estimate the anti-majoritarian tendency. The heuristic that simply counts the number of opinion updates in which the chosen opinion is the same as the minority opinion gives only a very poor approximation. These results show that the model learned by the proposed algorithm can be used to predict the future opinion share and provides a way to analyze such problems as influence maximization or minimization for opinion diffusion under the presence of anti-majoritarians.

The paper is organized as follows. We introduce the basic voter and anti-voter models in Section 2 and our proposed models, VwV and VwMV models in Section 3. We then perform the behavior analysis for share prediction using the mean field theory and discuss the behavior qualitatively in Section 4, and describe the parameter learning algorithm in Section 5. We detail the results of experimental evaluations in Section 6. We summarize what has been achieved and conclude the paper in Section 7.

2 Voter Models

We consider the diffusion of opinions in a social network represented by an undirected (bidirectional) graph $G = (V, E)$ with self-loops, where V and E ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. For a node $v \in V$, let $\Gamma(v)$ denote the set of neighbors of v in G, *i.e.*,

$$\Gamma(v) = \{u \in V;\ (u, v) \in E\}.$$

Note that $v \in \Gamma(v)$. We revisit the basic voter model that is one of the standard models of opinion dynamics, and the anti-voter model that is its variant, where the number of opinions is set to two.

2.1 Basic Voter Model

According to the work of Even-Dar and Shapira (2007), we recall the definition of the basic voter model on network G. In the model, each node of G is endowed with two states; opinions 1 and 2. The opinions are initially assigned to all the nodes in G, and the evolution process unfolds in discrete time-steps $t = 1, 2, 3, \cdots$ as follows: At each time-step t, each node v picks a random neighbor u and adopts the opinion that u holds at time-step $t - 1$.

More formally, let $f_t : V \to \{1, 2\}$ denote the *opinion distribution* at time-step t, where $f_t(v)$ stands for the opinion of node v at time-step t. Then, $f_0 : V \to \{1, 2\}$ is the initial opinion distribution, and $f_t : V \to \{1, 2\}$ is inductively defined as follows: For any $v \in V$, node v selects its opinion according to the probability distribution,

$$P(f_t(v) = 1) = \frac{N_1(t-1, v)}{|\Gamma(v)|}$$

$$P(f_t(v) = 2) = \frac{N_2(t-1, v)}{|\Gamma(v)|}$$

where $N_k(t, v)$ is the number of v's neighbors that hold opinion k at time-step t for $k = 1, 2$.

2.2 Anti-voter Model

In the basic voter model, it is assumed that people tend to follow their neighbors' majority opinion. However, since it is a common phenomenon that there are always people who do not agree with the majority and support the minority opinion, the *anti-voter model* is defined and investigated (Huber and Reinert, 2004; Röllin, 2007; Donnelly and Welsh, 1984; Matloff, 1977). In the anti-voter model, the opinion evolution process is replaced as follows: At each time-step t, each node v picks a random neighbor u and changes its opinion to the opposite of the opinion that u holds at time-step $t - 1$, *i.e.*, node v selects its opinion according to the probability distribution,

$$P(f_t(v) = 1) = \frac{N_2(t-1, v)}{|\Gamma(v)|}$$

$$P(f_t(v) = 2) = \frac{N_1(t-1, v)}{|\Gamma(v)|}$$

We note that each individual tends to adopt the minority opinion among its neighbors instead.

3 Proposed Model

3.1 Value-weighted Voter Model

We extend the basic voter model to the *value-weighted voter (VwV) model* for our purpose. In the VwV model, the total number of opinions is set to K (≥ 2), and each node of G is endowed with $(K+1)$ states; opinions $1, \cdots, K$, and *neutral* (*i.e.*, no-opinion state). We consider that a node is *active* when it holds an opinion k, and a node is *inactive* when it does not have any opinion (*i.e.*, when its state is neutral). We assume that nodes never switch their states from active to inactive. In order to discuss the competitive diffusion of K opinions, we introduce the parameter w_k (> 0) for each opinion k, which is referred to as the *opinion value* of opinion k. In the same way as the basic voter model, let $f_t : V \to \{0, 1, 2, \cdots, K\}$ denote the opinion distribution at time t, where opinion 0 denotes the neutral state. Here, f_t is defined for any non-negative real number t since the VwV model incorporates time delay in an asynchronous way, *i.e.*, t is continuous. For any $t > 0$, let $\varphi_t(v)$ denote the latest opinion of node v (before time t), and let $n_k(t, v)$ denote the number of v's neighbors that hold opinion k as the latest opinion (before time t), *i.e.*,

$$n_k(t, v) = |\{u \in \Gamma(v);\ \varphi_t(u) = k\}|.$$

We define the evolution process of the VwV model. At the initial time $t = 0$, each opinion is assigned to only one node and all other nodes are in the neutral state. [1] Given a target time T, the evolution process unfolds in the following steps:

1. Each node v independently decides the next update time t' at its update time t according to some probability distribution such as an exponential distribution with parameter $\eta_v = 1$, [2] where the first update time is $t = 0$ for every node.
2. At update time t, the node v selects its opinion according to the probability distribution,

$$P(f_t(v) = k) \ = \ p_k(t, v, \boldsymbol{w}), \quad (k = 1, \cdots, K), \tag{1}$$

where $\boldsymbol{w} = (w_1, \cdots, w_K)$ and

$$p_k(t, v, \boldsymbol{w}) = \frac{w_k\, n_k(t, v)}{\sum_{j=1}^{K} w_j\, n_j(t, v)}, \quad (k = 1, \cdots, K). \tag{2}$$

3. The process is repeated from the initial time $t = 0$ until the next update-time passes a given final-time T.

Note that the basic voter model with K opinions is derived from the VwV model with uniform opinion values $w_1 = \cdots = w_K$.

[1] This may look a rather unnatural assumption because it is unlikely that all the different opinions are initiated at the same time. Since each opinion is initiated by a single person and the goal is to see how it is propagated, it should be allowed that each opinion is assigned to only one node and all the remaining nodes are in neutral states, *i.e.*, unaffected by any opinion yet. We could have changed the timing of each opinion's initial utterance, but chose the simplest case.

[2] This assumes that the average delay time is 1.

3.2 Value-weighted Mixture Voter Model

Since the anti-voter model aims to represent the phenomenon that people tend to follow their neighbors' minority opinion, the anti-voter model with K opinions can be defined by replacing Eq. (1) of the VwV model with

$$P(f_t(v) = k) \;=\; \frac{1}{K-1}\left(1 - \frac{n_k(t,v)}{\sum_{j=1}^{K} n_j(t,v)}\right), \quad (k = 1, \cdots, K).$$

Therefore, we can also extend the anti-voter model with K opinions to the *value-weighted anti-voter model* by replacing Eq. (1) with

$$P(f_t(v) = k) \;=\; \frac{1 - p_k(t,v,\boldsymbol{w})}{K-1}, \quad (k = 1, \cdots, K).$$

For our purpose, we extend the VwV model and define the *value-weighted mixture voter (VwMV) model* by replacing Eq. (1) with

$$P(f_t(v) = k) \;=\; (1-\alpha_v)\, p_k(t,v,\boldsymbol{w}) + \alpha_v \frac{1 - p_k(t,v,\boldsymbol{w})}{K-1}, \quad (k = 1, \cdots, K), \tag{3}$$

where α_v is a parameter with $0 \leq \alpha_v \leq 1$. Note that each individual located at node v tends to behave like a majoritarian if the value of α_v is small, and tends to behave like an anti-majoritarian if the value of α_v is large. Therefore, we refer to α_v as the *anti-majoritarian tendency* of node v.

4 Behavior Analysis

In what follows, we first mathematically define the share prediction problem in Subsection 4.1 and explain why it is important to use a model to predict the future. Then, in Subsection 4.2 we introduce the mean field approach which is a method used in statistical physics to analyze the average behavior of a complex dynamic system. We first apply this theory to analyze the VwV model in Subsection 4.3 and discuss its asymptotic behavior and the time needed to reach consensus. We then apply this theory to analyze the VwMV model in Subsection 4.4 and discuss its asymptotic behavior in a similar way. These theoretical analysis sheds a light on the opinion formation dynamics and makes the behavior easy to understand.

4.1 Share Prediction Problem

Based on our opinion dynamics model (the VwMV model), we investigate the problem of predicting how large a share each opinion will have at a future target time T when the opinion diffusion is observed from $t = 0$ to $t = T_0$ ($< T$). Let $\mathcal{D}_{T_0}$ be the observed opinion diffusion data in time-interval $[0, T_0]$, where $\mathcal{D}_{T_0}$ consists of a sequence of (v, t, k) such that node v changed its opinion to opinion k at time t for $0 \leq t \leq T_0$. For any opinion k, let $h_k(t)$ denote its *population* at time t, *i.e.*,

$$h_k(t) = |\{v \in V;\; f_t(v) = k\}|.$$

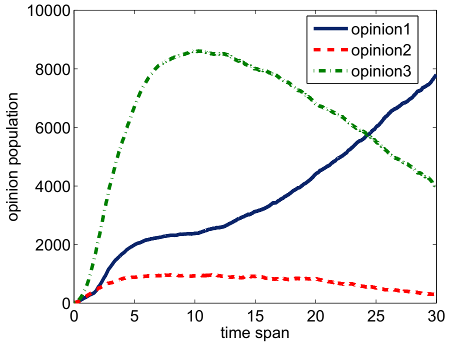


Fig. 1: An example of opinion population curves in the Blog network for $K = 3$.

Figure 1 shows an example of opinion population curves $h_1(t)$, $h_2(t)$, $h_3(t)$ for $K = 3$ in the Blog network (see Section 6 below), where the opinion values are set to $w_1 = 1.5$, $w_2 = 1.0$, $w_3 = 1.1$ and anti-majoritarian tendency α_v $(v \in V)$ is drawn from the beta distribution with shape parameters $a = 1$ and $b = 99$. Here, if we set $T_0 = 10$ and $T = 30$, we are able to observe $\mathcal{D}_{10}$ and thus $\{h_k(t);\, 0 \le t \le 10\}$ for $k = 1,2,3$ and the problem is to predict $h_1(30)$, $h_2(30)$, $h_3(30)$. Note that although the opinion dynamics is stochastic, we found that the variance of the value of $h_k(30)$ $(k = 1,2,3)$ is relatively small for $T_0 = 10$. We can easily see from Figure 1 that the naive time-series analysis method or a simple extrapolation method does not work well for this prediction problem. Thus, it is crucial to accurately estimate the values of the parameters of the VwMV model from the observed opinion diffusion data (more to come later on this).

Since the VwMV model gives a stochastic process, we introduce the *expected share* $g_k(t)$ of each opinion k at time t by

$$g_k(t) = \left\langle \frac{h_k(t)}{\sum_{j=1}^{K} h_j(t)} \right\rangle,$$

and consider the problem of predicting $g_k(t)$ $(k = 1, \cdots, K)$ from the observed data $\mathcal{D}_{T_0}$, which is referred to as the *share prediction problem*. Here, $\langle x \rangle$ denotes the expected value of a random variable x. For solving the share prediction problem, we develop a method that effectively estimates the values of the parameters w_k $(k = 1, \cdots, K)$ and α_v $(v \in V)$ from $\mathcal{D}_{T_0}$. We note that the method developed can also apply to detecting high anti-majoritarian tendency nodes (*i.e.*, anti-majoritarians) from the observed opinion diffusion data.

4.2 Mean Field Approach

Below, we theoretically investigate the asymptotic behavior of expected share $g_k(t)$ $(k = 1, \cdots, K)$ of the VwMV model for a sufficiently large t, and demonstrate that it is crucial to accurately estimate the values of the parameters, w_k, $(k = 1, \cdots, K)$ and α which is the average of α_v over all nodes $v \in V$.

According to previous work in statistical physics, (*e.g.*, Sood and Redner (2005)), we employ a mean field approach. We first consider a rate equation,

$$\frac{dg_k(t)}{dt} = (1 - g_k(t))\, P_k(t) - g_k(t)\,(1 - P_k(t)), \quad (k = 1, \cdots, K), \tag{4}$$

where $P_k(t)$ denotes the probability that a node adopts opinion k at time t. Note that in the right-hand side of Eq. (4), $g_k(t)$ is regarded as the probability of choosing a node holding opinion k at time t. Here, we assume that the average local opinion share,

$$\left\langle \frac{n_k(t,v)}{\sum_{j=1}^{K} n_j(t,v)} \right\rangle$$

in the neighborhood of a node v can be approximated by the expected opinion share $g_k(t)$ of the whole network for each opinion k. This assumption does not hold in general except that the network is a complete graph where every node's neighbors are all the nodes in the graph, which is not the case here. In fact, without this assumption, we cannot apply the mean field theory and analyze the average behavior of opinion dynamics. Extent to which this assumption is justified must await experimental evaluation by using the real network structure. As shown later, the assumption turned out to be acceptable. Under this assumption, we obtain the following approximation from Eq. (3):

$$P_k(t) \approx (1-\alpha)\,\tilde{p}_k(t,\boldsymbol{w}) + \alpha\,\frac{1-\tilde{p}_k(t,\boldsymbol{w})}{K-1}, \quad (k = 1,\cdots,K), \tag{5}$$

where α is the average value of anti-majoritarian tendency α_v, $(v \in V)$, and

$$\tilde{p}_k(t,\boldsymbol{w}) = \frac{w_k\, g_k(t)}{\sum_{j=1}^{K} w_j\, g_j(t)}, \quad (k = 1,\cdots,K). \tag{6}$$

Note that Eq. (5) is exactly satisfied when G is a complete network and the anti-majoritarian tendency is node independent, *i.e.*, $\alpha_v = \alpha$, $(\forall v \in V)$.

4.3 Analysis of VwV Model

For simplicity, we begin with the analysis of the VwV model. In this case, note that $\alpha_v = 0$ $(v \in V)$, *i.e.*, $\alpha = 0$.

4.3.1 Share Analysis

We analyze the behavior of expected share $g_k(t)$ $(k = 1,\cdots,K)$ of the VwV model for a sufficiently large t according to the above mean field approach. From Eqs. (4), (5) and (6), we have

$$\begin{aligned}\frac{dg_k(t)}{dt} &= (1-g_k(t))\frac{g_k(t)w_k}{\sum_{k'=1}^{K} g_{k'}(t)w_{k'}} - g_k(t)\left(1 - \frac{g_k(t)w_k}{\sum_{k'=1}^{K} g_{k'}(t)w_{k'}}\right) \\ &= \frac{g_k(t)w_k}{\sum_{k'=1}^{K} g_{k'}(t)w_{k'}} - g_k(t). \end{aligned} \tag{7}$$

Suppose that the opinion values are non-uniform, and let k^* be the opinion with the highest value parameter such that $w_{k^*} > w_k$ for all the other opinion k $(k \neq k^*)$. Here note that $w_k/w_{k^*} < 1$ for $k \neq k^*$. Then, we can obtain the following inequality from Eq. (7) when $g_k(t) > 0$ for all k:

$$\begin{aligned} \frac{dg_{k^*}(t)}{dt} &= \frac{g_{k^*}(t)w_{k^*}}{\sum_{k=1}^{K} g_k(t)w_k}\left(1-\sum_{k=1}^{K} g_k(t)\frac{w_k}{w_{k^*}}\right) \\ &> \frac{g_{k^*}(t)w_{k^*}}{\sum_{k=1}^{K} g_k(t)w_k}\left(1-\sum_{k=1}^{K} g_k(t)\right) = 0. \end{aligned}$$

Thus, unless $g_{k^*}(t) = 0$, the opinion k^* is expected to finally prevail the others, regardless of its current share since the function $g_{k^*}(t)$ is expected to increase as time passes until each of the other opinion shares becomes 0.

On the other hand, suppose that the opinion values are uniform (*i.e.*, $w_1 = \cdots = w_K$). Then, we obtain from Eq. (7) that

$$\frac{dg_k(t)}{dt} = 0, \quad (k = 1, \cdots, K).$$

Thus, if there exists some $t_0 > 0$ such that $g_1(t_0) = \cdots = g_K(t_0) = 1/K$, then $g_k(t) = 1/K$, $(t \geq t_0)$ for every opinion k. This implies that any opinion can in general become the majority.

Hence, we have the following results:

1. When the opinion values are uniform (*i.e.*, $w_1 = \cdots = w_K$), any opinion can become a winner.
2. When the opinion values are non-uniform, the opinion k^* with highest opinion value is expected to finally prevail over the others, that is, $\lim_{t\to\infty} g_{k^*}(t) = 1$.

These results suggest that it is crucially important to accurately estimate the opinion values of the VwV model from the observed data $\mathcal{D}_{T_0}$, [3] and imply that the share prediction problem can be well-defined only when the opinion values are non-uniform. We experimentally confirmed the results for several realistic networks, although the above analysis is valid only when the approximation (see Eq. (5)) holds.

4.3.2 Consensus Time Analysis

We further analyze the consensus time of the VwV model by using the above mean field approach when opinion values are non-uniform. For simplicity, we assume that $w_k = w$ if $k \neq k^*$, *i.e.*, the opinion values of the other opinions are the same. [4] Let r be the ratio of the value parameters defined by $r = w/w_{k^*}$. Then, we obtain the following differential equation for $g_{k^*}(t)$ from Eq. (7):

$$\begin{aligned} \frac{dg_{k^*}(t)}{dt} &= \frac{g_{k^*}(t)}{r(1-g_{k^*}(t))+g_{k^*}(t)} - g_{k^*}(t) \\ &= \frac{(1-r)g_{k^*}(t)(1-g_{k^*}(t))}{r+(1-r)g_{k^*}(t)}. \end{aligned}$$

From this differential equation, we can easily derive the following solution:

$$\frac{r}{1-r}\log(g_{k^*}(t)) - \frac{1}{1-r}\log(1-g_{k^*}(t)) = t + C,$$

[3] If the goal is to predict which opinion wins eventually, it is sufficient to identify which opinion has the highest value, but if we want to estimate the share of each opinion, we need to estimate the values accurately.

[4] This makes the analysis drastically simpler, but the results remains valid qualitatively.

where C stands for a constant of integration. Figure 2 shows examples of expected share curves based on the above solution with different ratios of the opinion values, where the ratio r is set to $r = 1-2^{-d}$ $(d = 1,2,3,4,5)$, and each curve is plotted from $t = 0$ by assuming $g_{k^*}(0) = 0.01$ until $t = T$ that satisfies $g_{k^*}(T) = 0.99$. From Figure 2, we can see that the consensus time is quite short when the ratio r is small, while it takes somewhat longer when the ratio r approaches to 1. More importantly, this result indicates that the consensus time of the VwV model is extremely short even when the ratio r is close to 1, compared with the basic voter model studied in previous work (*e.g.*, Even-Dar and Shapira (2007)). [5] Therefore, we consider that voter model can become more practical by introducing the opinion values.

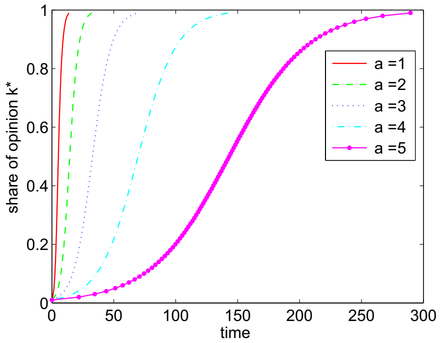


Fig. 2: Examples of expected share curves.

4.4 Analysis of VwMV model

Next, we analyze the behavior of expected share $g_k(t)$ $(k = 1,\cdots,K)$ of the VwMV model for a sufficiently large t according to the above mean field approach.

4.4.1 *Case of uniform opinion values:*

We suppose that $w_1 = \cdots = w_K$. Then, since $\sum_{k=1}^{K} g_k(t) = 1$, from Eq. (6), we obtain

$$\tilde{p}_k(t,\mathbf{w}) = g_k(t), \quad (k = 1,\cdots,K).$$

Thus, we can easily derive from Eqs. (4) and (5) that

$$\frac{dg_k(t)}{dt} = -\frac{\alpha}{1-1/K}\left(g_k(t) - \frac{1}{K}\right), \quad (k = 1,\cdots,K).$$

Hence, we have

$$\lim_{t\to\infty} g_k(t) = 1/K, \quad (k = 1,\cdots,K).$$

[5] Their results is that the basic voter model converges after $O(n^3 \log n)$ steps with probability 1- $o(1)$ where n is the number of nodes.

4.4.2 *Case of non-uniform opinion values:*

We assume that the opinion values are non-uniform. We parameterize the non-uniformity by the ratio,

$$s_k = \frac{w_k}{\sum_{j=1}^K w_j / K}, \quad (k = 1, \cdots, K).$$

Let k^* be the opinion with the highest opinion value. Note that $s_{k^*} > 1$. We assume as before for simplicity that

$$w_k = w (< w_{k^*}) \quad \text{if } k \neq k^*,$$

where w is a positive constant. We also assume that there exists some $t_0 > 0$ such that

$$g_1(t_0) = \cdots = g_K(t_0) = 1/K.$$

We can see from the symmetry of the setting that $g_k(t) = g_\ell(t)$, $(t \geq t_0)$ if k, $\ell \neq k^*$. This implies that opinion k^* is the winner at time t if and only if $g_{k^*}(t) > 1/K$. Then, from Eqs. (4) and (6), we obtain

$$\left.\frac{dg_{k^*}(t)}{dt}\right|_{t=t_0} = P_{k*}(t_0) - \frac{1}{K}, \qquad \tilde{p}_{k^*}(t_0, \boldsymbol{w}) = \frac{s_{k^*}}{K}.$$

Thus we have from Eq. (5) that

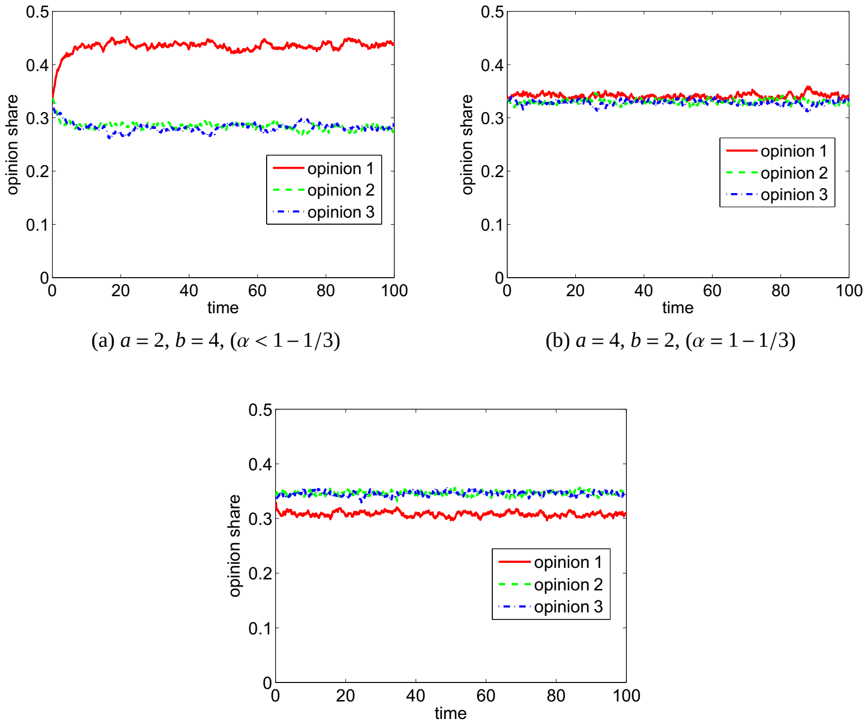
$$\left.\frac{dg_{k^*}(t)}{dt}\right|_{t=t_0} = \frac{s_{k^*} - 1}{K - 1}\left(1 - \frac{1}{K} - \alpha\right).$$

Therefore, we obtain the following results:

1. When $\alpha < 1 - 1/K$,
$$g_{k^*}(t) > 1/K, \quad (t > t_0),$$
that is, opinion k^* is expected to spread most widely and become the majority.
2. When $\alpha = 1 - 1/K$,
$$g_k(t) = 1/K, \quad (t \geq t_0),$$
for any opinion k, that is, any opinion can become a winner.
3. When $\alpha > 1 - 1/K$,
$$g_{k^*}(t) < 1/K, \quad (t > t_0),$$
that is, opinion k^* is expected to spread least widely and become the minority.

4.4.3 *Experiments:*

The above theoretical results are justified only when the approximation (see Eq. (5)) holds, which is always true in the case of complete networks. Real social networks are much more sparse and thus, we need to verify the extent to which the above results are true for real networks. We experimentally confirmed the above theoretical results for several real-world networks. Here, we present the experimental results for $K = 3$ in the Blog network (see Section 6), where the opinion values are $w_1 = 2$, $w_2 = w_3 = 1$, and anti-majoritarian tendency α_v, $(v \in V)$ is drawn from the beta distribution with certain combinations of shape parameters a and b. Figure 3 shows the results of opinion share curves, $t \mapsto h_k(t) \Big/ \sum_{j=1}^K h_j(t)$, $(k = 1, 2, 3)$,



(a) $a = 2$, $b = 4$, $(\alpha < 1 - 1/3)$

(b) $a = 4$, $b = 2$, $(\alpha = 1 - 1/3)$

(c) $a = 18$, $b = 2$, $(\alpha > 1 - 1/3)$

Fig. 3: Results of the opinion share curves for different distributions of anti-majoritarian tendency in the Blog network.

when the distribution of anti-majoritarian tendency changes, where each node adopted one of three opinions with equal probability at time $t = 0$. Note that

$$\begin{array}{ll} \alpha = 0.33 \ (< 1 - 1/3), & \text{if } a = 2, b = 4, \\ \alpha = 1 - 1/3, & \text{if } a = 4, b = 2, \\ \alpha = 0.9 \ (> 1 - 1/3), & \text{if } a = 18, b = 2. \end{array}$$

We obtained similar results to those in Figure 3 also for many other trials. These results support the validity of our theoretical analysis.

5 Learning Method

In this section we describe a method for estimating parameter values of the VwMV model from a given observed opinion spreading data $\mathcal{D}_{T_0}$. Based on the evolution process of our model (see Eq. (3)), we can obtain the likelihood function,

$$\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{w}, \alpha) = \log\left(\prod_{(v,t,k)\in\mathcal{D}_{T_0}} P(f_t(v) = k) \right), \qquad (8)$$

where $\boldsymbol{w}$ stands for the K-dimensional vector of opinion values, *i.e.*, $\boldsymbol{w} = (w_1, \cdots, w_K)$, and $\boldsymbol{\alpha}$ is the $|V|$-dimensional vector with each element α_v being the anti-majoritarian tendency of node v. Thus our estimation problem is formulated as a maximization problem of the objective function $\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{w}, \boldsymbol{\alpha})$ with respect to $\boldsymbol{w}$ and $\boldsymbol{\alpha}$. Note from Eqs. (2), (3) and (8) that $\mathcal{L}(\mathcal{D}_{T_0}; c\boldsymbol{w}, \boldsymbol{\alpha}) = c\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{w}, \boldsymbol{\alpha})$ for any $c > 0$. Note also that each opinion value w_k is positive. Thus, we transform the parameter vector $\boldsymbol{w}$ by $\boldsymbol{w} = \boldsymbol{w}(\boldsymbol{x})$, where

$$\boldsymbol{w}(\boldsymbol{x}) = (e^{x_1}, \cdots, e^{x_{K-1}}, 1), \quad \left(\boldsymbol{x} = (x_1, \cdots, x_{K-1}) \in \mathbf{R}^{K-1}\right). \tag{9}$$

Namely, our problem is to estimate the values of $\boldsymbol{x}$ and $\boldsymbol{\alpha}$ that maximize $\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{w}(\boldsymbol{x}), \boldsymbol{\alpha})$.

We derive an iterative algorithm for obtaining the maximum likelihood estimators. To this purpose, we introduce the following parameters that depend on $\boldsymbol{\alpha}$: For any $v \in V$ and $k, j \in \{1, \cdots, K\}$,

$$\beta_{v,k,j}(\boldsymbol{\alpha}) = \begin{cases} 1 - \alpha_v & \text{if } \; j = k, \\ \alpha_v/(K-1) & \text{if } \; j \neq k. \end{cases} \tag{10}$$

Then, from the definition of $P(f_t(v) = k)$ (see Eq. (3)), by noting $1 - p_k(t, v, \boldsymbol{w}) = \sum_{j \neq k} p_j(t, v, \boldsymbol{w})$, we can express Eq. (8) as follows:

$$\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{w}(\boldsymbol{x}), \boldsymbol{\alpha}) = \sum_{(v,t,k) \in \mathcal{D}_{T_0}} \log\left(\sum_{j=1}^{K} \beta_{v,k,j}(\boldsymbol{\alpha}) \; p_j(t, v, \boldsymbol{w}(\boldsymbol{x}))\right).$$

Now, let $\bar{\boldsymbol{z}}$ and $\bar{\boldsymbol{\alpha}}$ be the current estimates of $\boldsymbol{x}$ and $\boldsymbol{\alpha}$, respectively. Then, we define $q_{v,t,k,j}(\boldsymbol{x}, \boldsymbol{\alpha})$ by

$$q_{v,t,k,j}(\boldsymbol{x}, \boldsymbol{\alpha}) = \frac{\beta_{v,k,j}(\boldsymbol{\alpha}) \; p_j(t, v, \boldsymbol{w}(\boldsymbol{x}))}{\sum_{i=1}^{K} \beta_{v,k,i}(\boldsymbol{\alpha}) \; p_i(t, v, \boldsymbol{w}(\boldsymbol{x}))},$$

($v \in V$, $0 \leq t \leq T_0$, $k, j = 1, \cdots, K$), and transform our objective function as follows:

$$\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{w}(\boldsymbol{x}), \boldsymbol{\alpha}) = \mathcal{Q}(\boldsymbol{x}, \boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}) - \mathcal{H}(\boldsymbol{x}, \boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}), \tag{11}$$

where $\mathcal{Q}(\boldsymbol{x}, \boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}})$ is defined by

$$\mathcal{Q}(\boldsymbol{x}, \boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}) = \mathcal{Q}_1(\boldsymbol{x}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}) + \mathcal{Q}_2(\boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}), \tag{12}$$

$$\mathcal{Q}_1(\boldsymbol{x}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}) = \sum_{(v,t,k) \in \mathcal{D}_{T_0}} \sum_{j=1}^{K} q_{v,t,k,j}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}) \log p_j(t, v, \boldsymbol{w}(\boldsymbol{x})), \tag{13}$$

$$\mathcal{Q}_2(\boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}) = \sum_{(v,t,k) \in \mathcal{D}_{T_0}} \sum_{j=1}^{K} q_{v,t,k,j}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}) \log \beta_{v,k,j}(\boldsymbol{\alpha}), \tag{14}$$

and $\mathcal{H}(\boldsymbol{x}, \boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}})$ is defined by

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}) = \sum_{(v,t,k) \in \mathcal{D}_{T_0}} \sum_{j=1}^{K} q_{v,t,k,j}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}}) \log q_{v,t,k,j}(\boldsymbol{x}, \boldsymbol{\alpha}).$$

Since $\mathcal{H}(\boldsymbol{x}, \boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}})$ is maximized at $\boldsymbol{x} = \bar{\boldsymbol{x}}$ and $\boldsymbol{\alpha} = \bar{\boldsymbol{\alpha}}$, we can increase the value of $\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{w}(\boldsymbol{x}), \boldsymbol{\alpha})$ by maximizing $\mathcal{Q}(\boldsymbol{x}, \boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}})$ with respect to $\boldsymbol{x}$ and $\boldsymbol{\alpha}$ (see Eq. (11)). From Eq. (12), we can maximize $\mathcal{Q}(\boldsymbol{x}, \boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}})$ by independently maximizing $\mathcal{Q}_1(\boldsymbol{x}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}})$ and $\mathcal{Q}_2(\boldsymbol{\alpha}; \bar{\boldsymbol{x}}, \bar{\boldsymbol{\alpha}})$ with respect to $\boldsymbol{x}$ and $\boldsymbol{\alpha}$, respectively.

First, we estimate the value of $\boldsymbol{x}$ that maximizes $Q_1(\boldsymbol{x};\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})$. Here, note from Eqs.(2) and (9) that for $j = 1,\cdots,K$ and $\lambda = 1,\cdots,K-1$,

$$\frac{\partial p_j(t,v,\boldsymbol{w}(\boldsymbol{x}))}{\partial x_\lambda} = \delta_{j,\lambda}\, p_j(t,v,\boldsymbol{w}(\boldsymbol{x})) - p_j(t,v,\boldsymbol{w}(\boldsymbol{x}))\, p_\lambda(t,v,\boldsymbol{w}(\boldsymbol{x})), \tag{15}$$

where $\delta_{j,\lambda}$ is Kronecker's delta. From Eqs. (13) and (15), we have

$$\frac{\partial Q_1(\boldsymbol{x};\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})}{\partial x_\lambda} = \sum_{(v,t,k)\in\mathcal{D}_{T_0}} \sum_{j=1}^{K} q_{v,t,k,j}(\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}}) \left(\delta_{j,\lambda} - p_\lambda(t,v,\boldsymbol{w}(\boldsymbol{x}))\right), \tag{16}$$

for $\lambda = 1,\cdots,K-1$. Moreover, from Eqs. (15) and (16), we have

$$\frac{\partial^2 Q_1(\boldsymbol{x};\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})}{\partial x_\lambda\,\partial x_\mu} = \sum_{(v,t,k)\in\mathcal{D}_{T_0}} \sum_{j=1}^{K} q_{v,t,k,j}(\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}}) \Big(p_\lambda(t,v,\boldsymbol{w}(\boldsymbol{x}))\, p_\mu(t,v,\boldsymbol{w}(\boldsymbol{x})) - \delta_{\lambda,\mu}\, p_\lambda(t,v,\boldsymbol{w}(\boldsymbol{x}))\Big),$$

for $\lambda,\mu = 1,\cdots,K-1$. Thus, the Hessian matrix $(\partial^2 Q_1(\boldsymbol{x};\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})/\partial x_\lambda \partial x_\mu)$ is negative semi-definite since

$$\begin{aligned}
&\sum_{\lambda,\mu=1}^{K-1} \frac{\partial^2 Q_1(\boldsymbol{x};\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})}{\partial x_\lambda\,\partial x_\mu} y_\lambda y_\mu \\
&= \sum_{(v,t,k)\in\mathcal{D}_{T_0}} \sum_{j=1}^{K} q_{v,t,k,j}(\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}}) \left[\left(\sum_{\lambda=1}^{K-1} p_\lambda(t,v,\boldsymbol{w}(\boldsymbol{x}))y_\lambda\right)^2 - \sum_{\lambda=1}^{K-1} p_\lambda(t,v,\boldsymbol{w}(\boldsymbol{x}))y_\lambda{}^2\right] \\
&= -\sum_{(v,t,k)\in\mathcal{D}_{T_0}} \sum_{j=1}^{K} q_{v,t,k,j}(\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}}) \left[\sum_{\lambda=1}^{K-1} p_\lambda(t,v,\boldsymbol{w}(\boldsymbol{x})) \left(y_\lambda - \sum_{\mu=1}^{K-1} p_\mu(t,v,\boldsymbol{w}(\boldsymbol{x}))y_\mu\right)^2\right. \\
&\qquad\qquad \left. + \left(1 - \sum_{\lambda=1}^{K-1} p_\lambda(t,v,\boldsymbol{w}(\boldsymbol{x}))\right)\left(\sum_{\mu=1}^{K-1} p_\mu(t,v,\boldsymbol{w}(\boldsymbol{x}))y_\mu\right)^2\right] \\
&\leq 0,
\end{aligned} \tag{17}$$

for any $(y_1,\cdots,y_{K-1}) \in \mathbf{R}^{K-1}$. Hence, by solving the equations $\partial Q_1(\boldsymbol{x};\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})/\partial x_\lambda = 0$, $(\lambda = 1,\cdots,K-1)$ (see Eq. (16)), we can find the value of $\boldsymbol{x}$ that maximizes $Q_1(\boldsymbol{x};\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})$. We employed a standard Newton Method in our experiments.

Next, we estimate the value of $\boldsymbol{\alpha}$ that maximizes $Q_2(\boldsymbol{\alpha};\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})$. From Eqs. (10) and (14), we have

$$Q_2(\boldsymbol{\alpha};\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}}) = \sum_{(v,t,k)\in\mathcal{D}_{T_0}} \left(q_{v,t,k,k}(\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}}) \log(1-\alpha_v) + (1 - q_{v,t,k,k}(\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})) \log\left(\frac{\alpha_v}{K-1}\right)\right).$$

Note that $Q_2(\boldsymbol{\alpha};\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})$ is also a convex function of $\boldsymbol{\alpha}$. Therefore, we obtain the unique solution $\boldsymbol{\alpha}$ that maximizes $Q(\boldsymbol{x},\boldsymbol{\alpha};\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})$ as follows:

$$\alpha_v = \frac{1}{|\mathcal{D}_{T_0}(v)|} \sum_{(t,k)\in\mathcal{D}_{T_0}(v)} (1 - q_{v,t,k,k}(\bar{\boldsymbol{x}},\bar{\boldsymbol{\alpha}})),$$

for each $v \in V$, where $\mathcal{D}_{T_0}(v) = \{(t,k);\ (v,t,k) \in \mathcal{D}_{T_0}\}$.

When $\alpha_v = 0$ for any $v \in V$, the VwMV model is reduced to the VwV model. Thus, a straightforward application of the above learning algorithm for the VwMV model gives the learning algorithm for the VwV model. Note here that for the VwV model, the objective function becomes

$$\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{w}(\boldsymbol{x}), \boldsymbol{0}) = \sum_{(v,t,k) \in \mathcal{D}_{T_0}} \log p_k(t, v, \boldsymbol{w}(\boldsymbol{x})),$$

(see Eqs. (1) and (8)), and its second derivatives become

$$\frac{\partial^2 \mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{w}(\boldsymbol{x}), \boldsymbol{0})}{\partial x_\lambda \, \partial x_\mu} = \sum_{(v,t,k) \in \mathcal{D}_{T_0}} \Big(p_\lambda(t, v, \boldsymbol{w}(\boldsymbol{x}))\, p_\mu(t, v, \boldsymbol{w}(\boldsymbol{x})) - \delta_{\lambda,\mu}\, p_\lambda(t, v, \boldsymbol{w}(\boldsymbol{x})) \Big),$$

$(\lambda, \mu = 1, \cdots, K-1)$. In a similar way to Eq. (17), we can easily prove that the Hessian matrix $(\partial^2 \mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{w}(\boldsymbol{x}), \boldsymbol{0}) / \partial x_\lambda \, \partial x_\mu)$ is negative semi-definite. Therefore, we can guarantee that the optimal solution of the objective function is global optimal for the VwV model. Here, we mention that although it is not guaranteed that the optimal solution of the objective function of the VwMV model is global optimal, their estimated parameter values converged very closely to their true values in our experiments when there is an enough amount of training data.

6 Experimental Evaluation

Using large real networks, we experimentally investigate the capability of the proposed model and the performance of the proposed learning method. We first show the results of the accuracies of predicting future opinion shares. We then show the results of the estimation error of anti-majoritarian tendency, and the accuracies of detecting nodes with high anti-majoritarian tendency (*i.e.*, anti-majoritarians).

6.1 Experimental Settings

We employed four datasets of large real networks, which are all bidirectionally connected networks [6] and exhibit many of the key features of social networks. [7] The first one is a trackback network of Japanese blogs (Kimura et al, 2009) that has 12,047 nodes and 79,920 directed links (the Blog network). The second one is a Coauthor network (Palla et al, 2005) and has 12,357 nodes and 38,896 directed links (the Coauthor network). The third one is a network derived from the Enron Email Dataset (Klimt and Yang, 2004) by extracting the senders and the recipients and linking those that had bidirectional communications. It has 4,254 nodes and 44,314 directed links (the Enron network). The last one is a network of people that was derived from the "list of people" within Japanese Wikipedia (Kimura et al, 2009), which has 9,481 nodes and 245,044 directed links (the Wikipedia network). Just to provide a sense of how fast the opinion can propagate, the average shortest path of each network is given here: 8.175 for the Blog network, 8.160 for the Coauthor network, 3.726 for the Enron network and 4.700 for the Wikipedia network.

[6] Opinion propagation is directional. Choosing bidirectional networks means that opinion can propagate in both directions.

[7] It would be the best if we can use the real opinion propagation data. However, as we are not able to find such data, the next best is to use the network structures constructed from the real world social media data (not synthetic networks).

To do experiments, we have to first determine the values of parameters: the number of opinions K, the true value of each opinion w_k^*, the true value of each anti-majoritarian tendency α_v^*, $(v \in V)$. We varied $K = 2, 3, \cdots, 10$, and chose w_k^* from the interval $[0.5, 1.5]$ uniformly at random and α_v^* by drawing it from the beta distribution with the shape parameters a and b. We chose the beta distribution simply because of the easiness of controlling the average and variance of the distribution. As implied in Subsection 3.1, we used the exponential distribution with $\eta_v = 1$ to determine the opinion update time. Which nodes to start from is another problem. As explained also in Subsection 3.1 we assigned each opinion to only one node initially and all other nodes were set in the neutral state. Those initially assigned K nodes are taken from the top K nodes with respect to the node degree ranking. We start simulating the opinion propagation process from these K nodes using the parameter values which are assumed true, and generated $\mathcal{D}_{T_0}$. As for our learning settings, we set the initial value of each value parameter to $w_k = 1$, and the initial value of each anti-majoritarian tendency to $\alpha_v = 0.5$, $(v \in V)$. We terminated the learning iteration when the increase of our objective function becomes sufficiently small, *i.e.*,

$$\frac{\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{w}, \boldsymbol{\alpha}) - \mathcal{L}(\mathcal{D}_{T_0}; \bar{\boldsymbol{w}}, \bar{\boldsymbol{\alpha}})}{\mathcal{L}(\mathcal{D}_{T_0}; \bar{\boldsymbol{w}}, \bar{\boldsymbol{\alpha}})} < 10^{-8},$$

where $\boldsymbol{w}$ and $\boldsymbol{\alpha}$ mean the parameter vectors updated from $\bar{\boldsymbol{w}}$ and $\bar{\boldsymbol{\alpha}}$. Note that our learning algorithm always increases our objective function as described in the previous section.

6.2 Share Prediction

For each number of opinions $(k = 2, 3, \cdots, K)$ we predicted the expected share $g_k(T)$ for the observed data $\mathcal{D}_{T_0}$, where we set $T = 30$, and investigated the cases $T_0 = 10, 15$ and $\alpha = 0.5, 0.1, 0.01$ by generating α_v with $(a, b) = (2, 2), (1, 9), (1, 99)$, respectively. As we mentioned in Section 1 we think it is important to learn the model using a small amount of data and predict the near future. Since the average shortest path of each network is less than 10, $T_0 = 10$ is the minimum training time required to learn the parameters for all the nodes. Note that α means the average anti-majoritarian tendency, which is given by $a/(a+b)$. Namely, after we have estimated the values of each w_k and each α_v, we predicted the value of $g_k(T)$ by simulating the model M times from $\mathcal{D}_{T_0}$ and taking their average, where we used $M = 100$. In fact, our preliminary experiments indicate that the results for $M = 100$ are not much different from those for $M = 1,000$ and $10,000$ in the networks we used.

In order to investigate the importance of introducing the anti-majoritarian tendency of each node, we compared the proposed method with the VwV model which has no anti-majoritarian component. Moreover, in order to investigate the importance of introducing the opinion values, we also compared the proposed method with the same VwMV model in which the opinion values are constrained to take a uniform value and the anti-majoritarian tendency of each node is the only parameter to be estimated. We refer to this method as the *uniform value method*. Furthermore, given the observed data $\mathcal{D}_{T_0}$, we can simply apply a polynomial extrapolation for predicting the expected share of opinion k at a target time T, since we can naively speculate that the recent trend for each opinion captured by the polynomial function approximation continues. Thus, we consider predicting the values of $g_1(T)$, $\cdots$, $g_K(T)$, by estimating the value of the population $h_k(T)$ of opinion k at time T based on the polynomial function of degree L that interpolates the $L+1$ data points $\{(T_0 - \Delta + \ell\Delta/L, h_k(T_0 - \Delta + \ell\Delta/L)); \ell = 0, 1, \cdots, L\}$, where Δ is the parameter with $0 < \Delta \leq T_0$. We refer to this prediction method as the *polynomial extrapolation method*. In our experiments,

Table 1: Results of opinion share prediction for the Blog network ($T_0 = 10$, $\alpha = 0.1$, $K = 10$). Note that the two-side 0.05 point of the t-distribution with 9 degrees of freedom is 2.262.

Method	Average of error $\mathcal{E}_g$	t-value $\mathcal{T}_g^{PC}$
proposed	0.0396	—
VwV	0.4520	15.4645
uniform value	0.5172	13.4893
linear ($\varDelta = 1$)	0.4996	12.5024
linear ($\varDelta = 3$)	0.4243	12.0177
linear ($\varDelta = 5$)	0.3247	14.4648
quadratic ($\varDelta = 1$)	1.0845	13.2210
quadratic ($\varDelta = 3$)	1.2795	26.9768
quadratic ($\varDelta = 5$)	1.3296	15.6869
cubic ($\varDelta = 1$)	1.3710	18.3478
cubic ($\varDelta = 3$)	1.1799	12.5790
cubic ($\varDelta = 5$)	1.1219	16.7674
quartic ($\varDelta = 1$)	1.1963	19.0506
quartic ($\varDelta = 3$)	1.1079	16.2728
quartic ($\varDelta = 5$)	1.0956	11.9049

Table 2: Results of opinion share prediction for the Coauthor network ($T_0 = 10$, $\alpha = 0.1$, $K = 10$). Note that the two-side 0.05 point of the t-distribution with 9 degrees of freedom is 2.262.

Method	Average of error $\mathcal{E}_g$	t-value $\mathcal{T}_g^{PC}$
proposed	0.0590	—
VwV	0.4634	15.9587
uniform value	0.4422	13.5598
linear ($\varDelta = 1$)	0.4193	14.1568
linear ($\varDelta = 3$)	0.2814	10.2062
linear ($\varDelta = 5$)	0.2097	9.5952
quadratic ($\varDelta = 1$)	1.0794	17.6792
quadratic ($\varDelta = 3$)	1.2158	12.9942
quadratic ($\varDelta = 5$)	1.6140	22.1016
cubic ($\varDelta = 1$)	1.1616	16.4268
cubic ($\varDelta = 3$)	1.1615	17.8509
cubic ($\varDelta = 5$)	0.9575	18.6748
quartic ($\varDelta = 1$)	1.1852	14.3082
quartic ($\varDelta = 3$)	1.0971	14.1797
quartic ($\varDelta = 5$)	1.1889	17.3193

we adopted $L = 1, 2, 3, 4$, *i.e.*, the linear, quadratic, cubic, and quartic polynomial functions, and examined $\varDelta = 1$, $\varDelta = 3$, and $\varDelta = 5$. We evaluated the effectiveness of the proposed share prediction method by comparing it with the above six methods (VwV, uniform and four polynomial).

Let $\widehat{g}_k(T)$ be the estimate of $g_k(T)$ by a share prediction method. We measured the performance of the share prediction method by the prediction error $\mathcal{E}_g$ defined by [8]

$$\mathcal{E}_g = \sum_{k=1}^{K} |\widehat{g}_k(T) - g_k(T)|.$$

[8] It may sound more reasonable to weight each difference by the share itself, but we decided not to do so. We rather considered the prediction problem as the classification problem.

Table 3: Results of opinion share prediction for the Enron network ($T_0 = 10$, $\alpha = 0.1$, $K = 10$). Note that the two-side 0.05 point of the t-distribution with 9 degrees of freedom is 2.262.

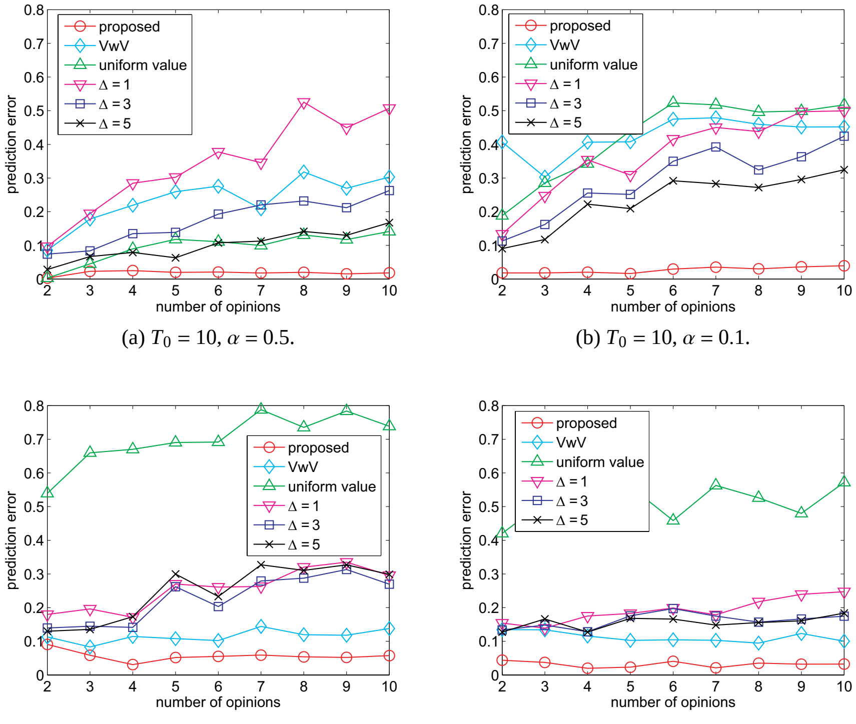
Method	Average of error $\mathcal{E}_g$	t-value $\mathcal{T}_g^{PC}$
proposed	0.0731	—
VwV	0.6030	12.7367
uniform value	0.6088	20.4684
linear ($\Delta = 1$)	0.6909	8.4882
linear ($\Delta = 3$)	0.6511	11.2945
linear ($\Delta = 5$)	0.5577	15.1556
quadratic ($\Delta = 1$)	1.1341	11.4784
quadratic ($\Delta = 3$)	1.0765	13.9631
quadratic ($\Delta = 5$)	1.1763	14.7290
cubic ($\Delta = 1$)	1.2378	15.1644
cubic ($\Delta = 3$)	1.1378	16.2330
cubic ($\Delta = 5$)	1.2605	16.6612
quartic ($\Delta = 1$)	1.1699	11.4147
quartic ($\Delta = 3$)	1.3411	32.1862
quartic ($\Delta = 5$)	1.1910	15.0670

Table 4: Results of opinion share prediction for the Wikipedia network ($T_0 = 10$, $\alpha = 0.1$, $K = 10$). Note that the two-side 0.05 point of the t-distribution with 9 degrees of freedom is 2.262.

Method	Average of error $\mathcal{E}_g$	t-value $\mathcal{T}_g^{PC}$
proposed	0.0390	—
VwV	0.4429	12.8927
uniform value	0.6000	11.6327
linear ($\Delta = 1$)	0.5151	13.2910
linear ($\Delta = 3$)	0.4073	12.2377
linear ($\Delta = 5$)	0.3968	14.8808
quadratic ($\Delta = 1$)	1.1122	12.8117
quadratic ($\Delta = 3$)	1.1521	15.0864
quadratic ($\Delta = 5$)	1.1674	16.0370
cubic ($\Delta = 1$)	1.2193	13.7714
cubic ($\Delta = 3$)	1.1950	16.4728
cubic ($\Delta = 5$)	1.0156	16.7386
quartic ($\Delta = 1$)	1.0679	12.1467
quartic ($\Delta = 3$)	1.2045	18.3987
quartic ($\Delta = 5$)	1.3886	27.4023

We first examined the case of $T_0 = 10$, $\alpha = 0.1$ and $K = 10$. Tables 1, 2, 3 and 4 are the results of opinion share prediction for the Blog, the Coauthor, the Enron and the Wikipedia networks, respectively. We conducted 10 trials varying the true values of value parameters for each K, and the second column in Tables 1, 2, 3 and 4 indicates the average of $\mathcal{E}_g$ over the 10 trials. In order to investigate whether the difference of the prediction error $\mathcal{E}_g$ between the proposed method and each of the other methods used for comparison is statistically significant or not, we performed a t-test. Let $\mathcal{E}_g^P$ and $\mathcal{E}_g^C$ denote the values of $\mathcal{E}_g$ for the proposed method and the compared method, respectively. We calculated t-value

$$\mathcal{T}_g^{PC} = \frac{\sqrt{10}\,\mathrm{mean}\left(\mathcal{E}_g^P - \mathcal{E}_g^C\right)}{\mathrm{std}\left(\mathcal{E}_g^P - \mathcal{E}_g^C\right)},$$



(a) $T_0 = 10$, $\alpha = 0.5$.

(b) $T_0 = 10$, $\alpha = 0.1$.

(c) $T_0 = 10$, $\alpha = 0.01$.

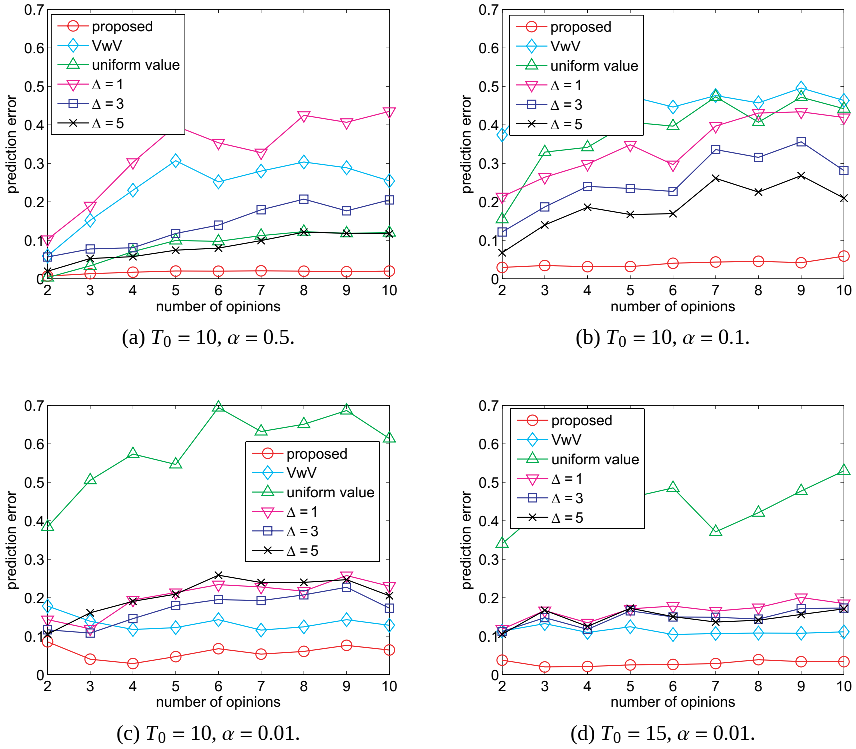
(d) $T_0 = 15$, $\alpha = 0.01$.

Fig. 4: Results of opinion share prediction for the Blog network.

where mean(x) and std(x) denote the standard average and the sample standard deviation of sample x, respectively. In Tables 1, 2, 3 and 4, the third column indicates the t-value $\mathcal{T}_g^{PC}$. Here, note that the two-side 0.05 point of the t-distribution with 9 degrees of freedom is $t^*_{9,0.05} = 2.262$. Thus, we see that in the case of $T_0 = 10$, $\alpha = 0.1$ and $K = 10$, the difference between the proposed method and each of the compared methods in prediction error $\mathcal{E}_g$ is statistically significant by the t-test at significance level 0.05. Moreover, from Tables 1, 2, 3 and 4, we see that the linear extrapolation method performed best among the polynomial extrapolation methods in the case of $T_0 = 10$, $\alpha = 0.1$ and $K = 10$. We obtained the same results for the other cases with different combinations of T_0, α and K. Thus, we show only the results of the linear extrapolation method for the polynomial extrapolation method.

Figure 4 is the results for the Blog network, where circles, diamonds and upward triangles indicate the prediction errors of the proposed method, the VwV method, and the uniform value method, respectively, and downward triangles, squares, and crosses indicate those of the linear extrapolation method adopting $\varDelta = 1$, $\varDelta = 3$, and $\varDelta = 5$, respectively. Figure 4 (a), (b), (c) and (d) are the results for $(T_0, \alpha) = (10, 0.5)$, $(10, 0.1)$, $(10, 0.01)$, and $(15, 0.01)$, respectively. Figures 5, 6, and 7 are the results for the other three networks, *i.e.*, the Coauthor network, the Enron network, and the Wikipedia network, respectively.

From these figures, we see that the proposed method worked substantially better than the other methods. More specifically, the VwV method worked poorly when values for the anti-majoritarian tendency were relatively large. Conversely, the uniform value method



(a) $T_0 = 10$, $\alpha = 0.5$.

(b) $T_0 = 10$, $\alpha = 0.1$.

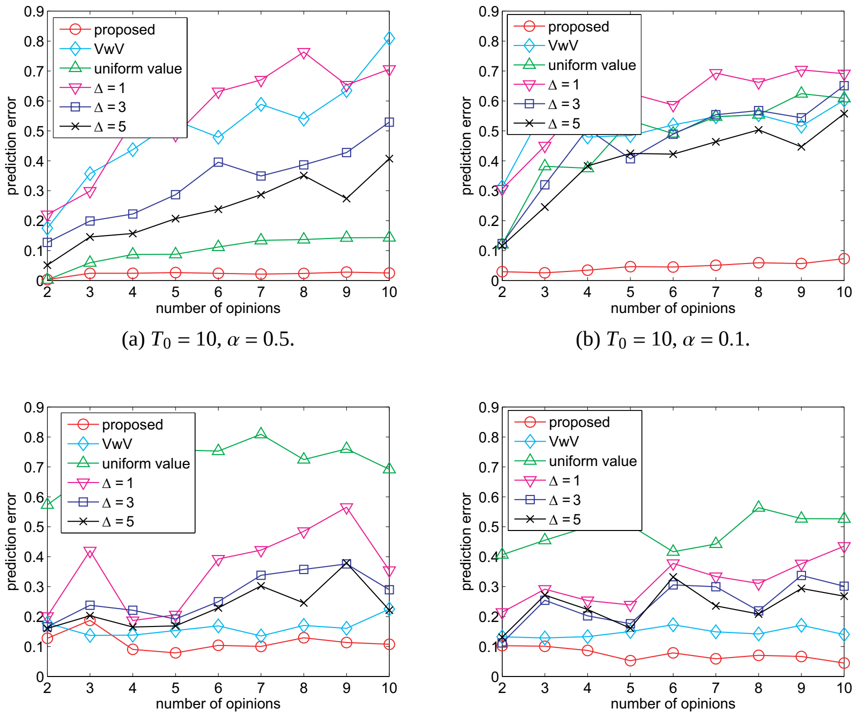
(c) $T_0 = 10$, $\alpha = 0.01$.

(d) $T_0 = 15$, $\alpha = 0.01$.

Fig. 5: Results of opinion share prediction for the Coauthor network

worked poorly when they were relatively small. These results are predictable because the VwV method cannot cope with the effect of the anti-majoritarian tendency and the uniform value method cannot cope with the effect of opinion value. We further see that the proposed method significantly outperformed the *polynomial extrapolation method* in every case. Especially, we observed that the proposed method accurately predicted the share at T even in the case that the share ranking at T_0 got reversed at the target time T as shown in Figure 1. This is attributed to the use of the estimated value parameters which take different values for different opinions, and is consistent with the results of the mean field analysis. We also observe that compared with cases of $\alpha = 0.5$ and $\alpha = 0.1$, the performance of the proposed method in case of $\alpha = 0.01$ becomes worse for $T_0 = 10$. This is because the opinion change driven by the anti-majoritarians is smaller when α is smaller, thereby providing less effective training data for learning α. Larger error for α negatively affects the results of share prediction despite the effect of anti-majoritarians is less. However, it becomes better and comparable to the other cases for $T_0 = 15$ as expected since the amount of training data increases.

During the experiments we noticed that the time needed to reach the consensus gets longer when the difference between the largest and the second largest values of the opinion value parameters is small. This can also be predicted by the consensus time analysis, *i.e.*, considering the case where the highest two values are the same and the rest are also the same.



(a) $T_0 = 10$, $\alpha = 0.5$.

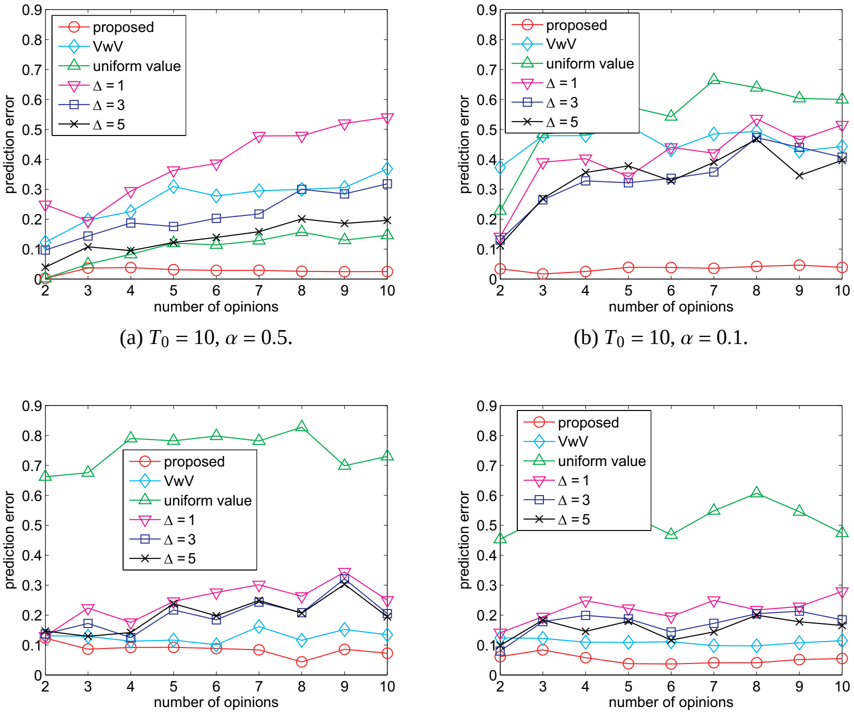
(b) $T_0 = 10$, $\alpha = 0.1$.

(c) $T_0 = 10$, $\alpha = 0.01$.

(d) $T_0 = 15$, $\alpha = 0.01$.

Fig. 6: Results of opinion share prediction for the Enron network

In this subsection, we focused only on the accuracy of share prediction and did not discuss the accuracy of parameter learning. As conjectured in Section 1, learning the opinion values is easy and learning the anti-majoritarian tendency is hard. Indeed, all the opinion values can be estimated in good accuracy. The average error was 6% even using a training data for such a short period of time. However, as predicted, the average error of the estimated anti-majoritarian tendency is large. For example, in the case of $T_0 = 10$, $\alpha = 0.5$ and $K = 10$, the average value of error $\mathcal{E}_\alpha$ was more than 0.17 for all the four networks. Namely, the estimation error of anti-majoritarian tendency for each node was more than $(0.17/0.5) * 100 = 34\%$ on the average. This is because the number of parameters is the same as the number of nodes which is very large. Nevertheless, the accuracy of share prediction is very good. For example, in the case of $T_0 = 10$, $\alpha = 0.5$ and $K = 10$, the average value of error $\mathcal{E}_g$ was less than 0.026 for all the four networks. Namely, the share prediction error for each opinion was less than $((0.026/10)/(1/10)) * 100 = 2.6\%$ on the average. This looks strange at a glance, but we can explain the reason as follows. We started with the K distinct initial nodes and all the other nodes were neutral in the beginning. Recall that we set the average time delay to 1.0, which means that on the average each node updates its opinion every single time unit. Thus when $T_0 = 10$ the opinion updates can propagate 10 steps on the average. As explained in Subsection 6.2, considering that the average shortest path of the network is less than 10 for all the networks, opinion update takes place barely almost all the nodes. For some nodes the number of updates is 10 and for other nodes it is 1. The accuracy



(a) $T_0 = 10$, $\alpha = 0.5$. (b) $T_0 = 10$, $\alpha = 0.1$.

(c) $T_0 = 10$, $\alpha = 0.01$. (d) $T_0 = 15$, $\alpha = 0.01$.

Fig. 7: Results of opinion share prediction for the Wikipedia network

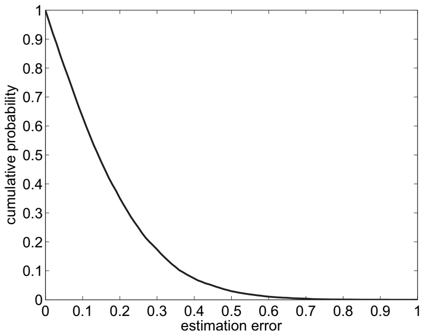


Fig. 8: Distribution of estimation error for anti-majority tendency of each node in the Blog network ($T_0 = 10$, $\alpha = 0.5$, $K = 10$).

of the anti-majoritarian tendency for these nodes where the opinion updates are very few is indeed very bad (no valid learning took place), but the accuracy for the nodes that undergo several opinion updates is good. The variance of the node-wise accuracy is large. Figure 8 which is the cumulative error probability $P(|\hat{\alpha}_v - \alpha_v^*| \geq x)$ in case of $T_0 = 10$, $\alpha = 0.5$ and $K = 10$ for the Blog network clearly indicates this, where each α_v^* and $\hat{\alpha}_v$ denote the true and the estimated anti-majoritarian tendencies of node v, respectively. The average error is indeed large and about 30% of nodes have errors greater than 50%. However, as the mean field analysis implies, it is the average of the anti-majoritarian tendency that matters, as the first approximation, as far as the opinion share is concerned. In this case, we can verify that $\sum_{v \in V} \left|\hat{\alpha}_v - \alpha^*_v\right| / |V| = 0.1811$ and $\left|\sum_{v \in V} \hat{\alpha}_v / |V| - \sum_{v \in V} \alpha^*_v / |V|\right| = 0.0015$. The latter is three orders of magnitude less. This explains the good accuracy of the opinion share despite the bad accuracy of the anti-majoritarian tendency. In the next subsection we will describe the accuracy of the anti-majoritarian tendency using more training data.

To sum up, we confirmed that the results of our theoretical analyses hold in these real networks and that the proposed method outperforms the *polynomial extrapolation method*. On the average, the prediction error of the proposed method was about four times less for a given T_0. Besides, it achieved a comparable prediction accuracy with the observation time three times less compared with the *polynomial extrapolation method*.

6.3 Discovery of Anti-majority Opinionists

We examined the accuracy of discovering anti-majoritarian opinionists (and majoritarian opinionists) for both a small ($K = 3$) and a large ($K = 10$) K, by varying $T_0 = 100, 200, \cdots,$ 1000. The error is measured by $\mathcal{E}_\alpha$,

$$\mathcal{E}_\alpha = \frac{1}{|V|} \sum_{v \in V} |\hat{\alpha}_v - \alpha^*_v|.$$

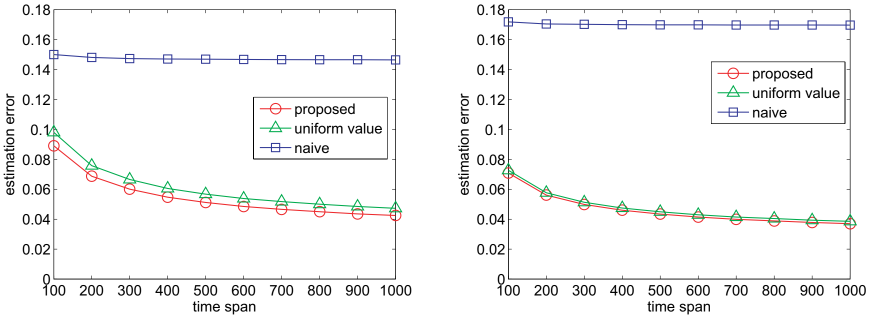
We also measured the accuracies of detecting the high and the low anti-majoritarian tendency nodes by F-measures $\mathcal{F}_A$ and $\mathcal{F}_N$, respectively. Here, $\mathcal{F}_A$ and $\mathcal{F}_N$ are defined as follows:

$$\mathcal{F}_A = \frac{2|\hat{A} \cap A^*|}{|\hat{A}| + |A^*|}, \quad \mathcal{F}_N = \frac{2|\hat{N} \cap N^*|}{|\hat{N}| + |N^*|},$$

where A^* and $\hat{A}$ are the sets of the true and the estimated top 15% nodes of high anti-majoritarian tendency, respectively, and N^* and $\hat{N}$ are the sets of the true and the estimated top 15% nodes of low anti-majoritarian tendency, respectively.

We compared the proposed method with the naive approach in which the anti-majoritarian tendency of a node is estimated by simply counting the number of opinion updates in which the opinion chosen by the node is the minority's opinion in its neighborhood. We refer to the method as the *naive counting method*. We also compared the proposed method with the uniform value method mentioned in the previous subsection.

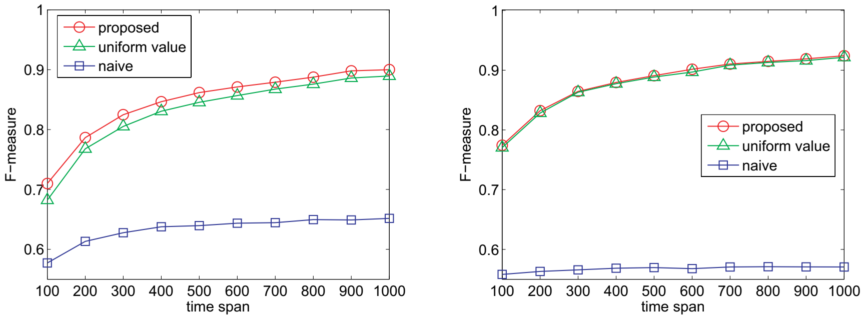
Figures 9 and 10 are the results for the Blog network, where circles, upward triangles, and squares indicate the prediction errors and the F-measure performance of the proposed method, the uniform value method, and *naive method*, respectively. Figures 9 (a), and (b) show the estimation error $\mathcal{E}_\alpha$ of each method as a function of time span T_0 with $K = 3$ and $K = 10$, respectively, while Figures 10 (a) and (b) the F-measure $\mathcal{F}_A$ of each method as a function of time span T_0 with $K = 3$ and $K = 10$, respectively. Here, we repeated the same experiment 10 times independently, and plotted the average over the 10 results. Figures 11, 12, 13, 14, 15 and 16 are the results for the other three networks, *i.e.*, the Coauthor network, the Enron network, and the Wikipedia network, respectively. Note that we only showed the results for $\alpha = 0.5$, *i.e.*, $a = b = 2$, because we obtained quite similar results for the other anti-majoritarian tendency α.



(a) Results for $K = 3$.

(b) Results for $K = 10$.

Fig. 9: Estimation errors of anti-majoritarian tendency for the Blog network.



(a) Results for $K = 3$.

(b) Results for $K = 10$.

Fig. 10: Accuracies of extracting nodes with high anti-majoritarian tendency for the Blog network.

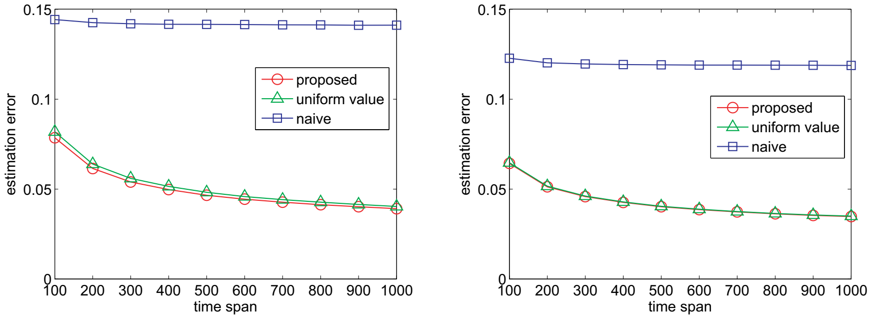
Table 5: Results for estimation errors of anti-majoritarian tendency for the Blog network ($T_0 = 1000$). Note that the two-side 0.05 point of the t-distribution with 9 degrees of freedom is $t^*_{9,0.05} = 2.262$.

Method	$K = 3$ Average of error $\mathcal{E}_\alpha$	$K = 3$ t-value $\mathcal{T}_\alpha^{PC}$	$K = 10$ Average of error $\mathcal{E}_\alpha$	$K = 10$ t-value $\mathcal{T}_\alpha^{PC}$
proposed	0.0229	—	0.0169	—
uniform value	0.0280	5.6016	0.0186	5.0033
naive	0.1403	229.4537	0.1607	577.3649

In order to investigate whether the difference between the proposed method and each of the other methods is statistically significant or not, we in particular performed a t-test for estimation error $\mathcal{E}_\alpha$. Let $\mathcal{E}_\alpha^P$ and $\mathcal{E}_\alpha^C$ denote the values of $\mathcal{E}_\alpha$ for the proposed method and a compared method, respectively. We calculated t-value

$$\mathcal{T}_\alpha^{PC} = \frac{\sqrt{10}\ \mathrm{mean}\left(\mathcal{E}_\alpha^P - \mathcal{E}_\alpha^C\right)}{\mathrm{std}\left(\mathcal{E}_\alpha^P - \mathcal{E}_\alpha^C\right)},$$

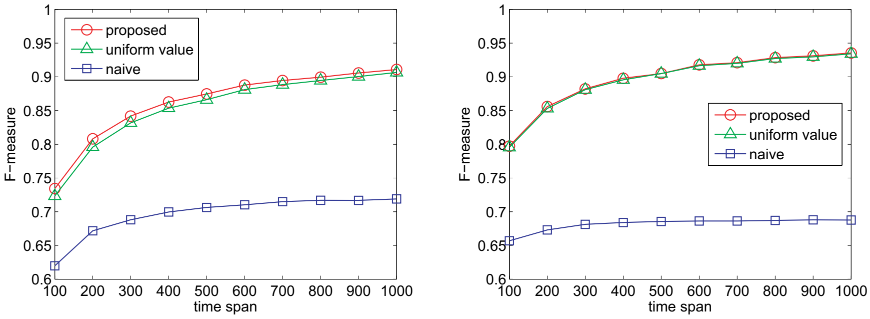
where mean(x) and std(x) are defined in the previous section. Tables 5, 6, 7, and 8 show the results for estimation errors of anti-majoritarian tendency in the case of $T_0 = 1000$ for the



(a) Results for $K = 3$.

(b) Results for $K = 10$.

Fig. 11: Estimation errors of anti-majoritarian tendency for the Coauthor network.



(a) Results for $K = 3$.

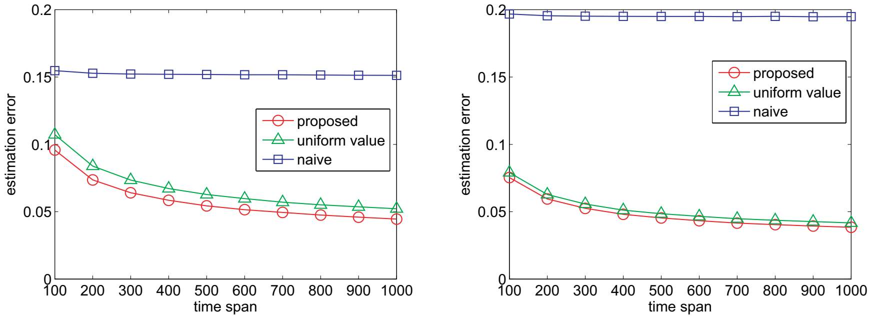
(b) Results for $K = 10$.

Fig. 12: Accuracies of extracting nodes with high anti-majoritarian tendency for the Coauthor network.

Table 6: Results for estimation errors of anti-majoritarian tendency for the Coauthor network ($T_0 = 1000$). Note that the two-side 0.05 point of the t-distribution with 9 degrees of freedom is $t^*_{9,0.05} = 2.262$.

Method	$K = 3$ Average of error $\mathcal{E}_\alpha$	$K = 3$ t-value $\mathcal{T}^{PC}_\alpha$	$K = 10$ Average of error $\mathcal{E}_\alpha$	$K = 10$ t-value $\mathcal{T}^{PC}_\alpha$
proposed	0.0195	—	0.0147	—
uniform value	0.0208	4.0840	0.0150	9.9920
naive	0.1350	404.8052	0.1074	526.8500

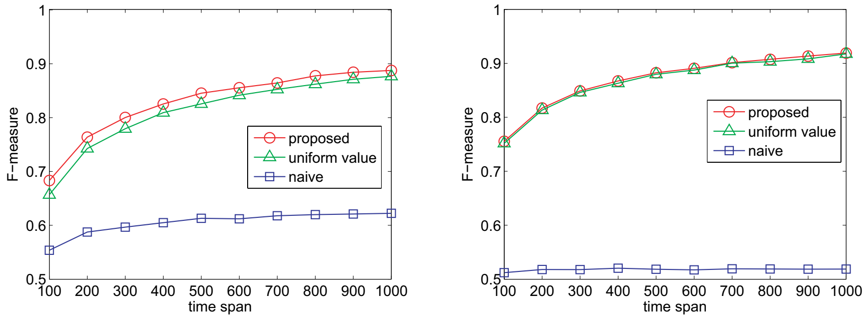
Blog, the Coauthor, the Enron, and the Wikipedia networks, respectively. Here, the second and the fourth columns indicate the average of $\mathcal{E}_\alpha$ over the 10 trials for the cases of $K = 3$ and $K = 10$, respectively. Also, the third and the fifth columns indicate t-value $\mathcal{T}^{PC}_\alpha$ for the cases of $K = 3$ and $K = 10$, respectively. Note that the two-side 0.05 point of the t-distribution with 9 degrees of freedom is $t^*_{9,0.05} = 2.262$. Thus, from Tables 5, 6, 7, and 8, we see that in the case of $T_0 = 1000$, the difference between the proposed and each comparison methods in prediction error $\mathcal{E}_\alpha$ is statistically significant by the t-test at significance level 0.05. Note that we only showed the results for $T_0 = 1000$, because we obtained quite similar results for other values of $T_0 \geq 100$. As explained in Subsection 6.2, $T_0 = 10$ is too short for learning anti-majoritarians.



(a) Results for $K = 3$.

(b) Results for $K = 10$.

Fig. 13: Estimation errors of anti-majoritarian tendency for the Enron network.



(a) Results for $K = 3$.

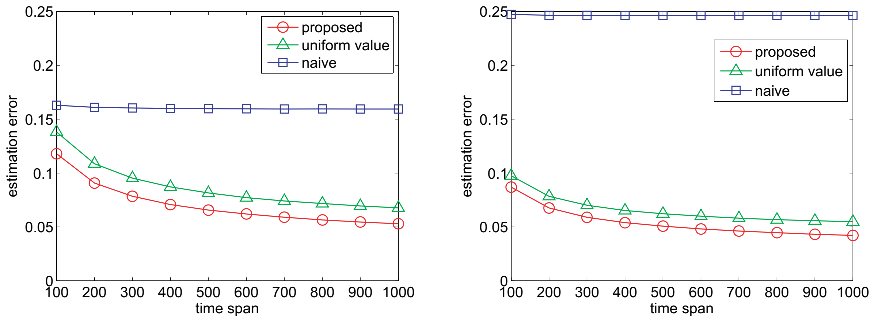
(b) Results for $K = 10$.

Fig. 14: Accuracies of extracting nodes with high anti-majoritarian tendency for the Enron network.

Table 7: Results for estimation errors of anti-majoritarian tendency for the Enron network ($T_0 = 1000$). Note that the two-side 0.05 point of the t-distribution with 9 degrees of freedom is $t^*_{9,0.05} = 2.262$.

	$K = 3$	$K = 3$	$K = 10$	$K = 10$
Method	Average of error $\mathcal{E}_\alpha$	t-value $\mathcal{T}^{PC}_\alpha$	Average of error $\mathcal{E}_\alpha$	t-value $\mathcal{T}^{PC}_\alpha$
proposed	0.0254	—	0.0186	—
uniform value	0.0331	3.8280	0.0220	8.5671
naive	0.1453	101.3125	0.1863	306.6563

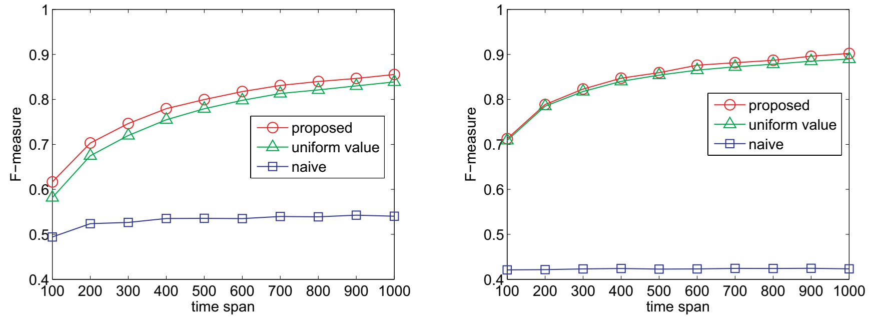
As expected, $\mathcal{E}_\alpha$ decreases, and $\mathcal{F}_A$ increases as T_0 increases (*i.e.*, the amount of training data $\mathcal{D}_{T_0}$ increases). We observe that the proposed method performs the best, the uniform value method follows, and the naive method behaves very poorly for all the networks. Here, we note that quite similar results were also observed for $\mathcal{F}_N$, *i.e.*, extracting nodes with low anti-majoritarian tendency although those results are not reported in this paper. The proposed method can detect both the anti-majoritarians and the majoritarians with the accuracy greater than 90% at $T = 1000$ for all cases. We can also see that the proposed method is not sensitive to both K and the network structure because of the explicit use of the model, but the other two methods are so. For example, although the uniform value method of $K = 10$ performs well in $\mathcal{F}_A$ for the Blog, Coauthor and Enron networks, it does not so in $\mathcal{F}_A$ for the Wikipedia



(a) Results for $K = 3$.

(b) Results for $K = 10$.

Fig. 15: Estimation errors of anti-majoritarian tendency for the Wikipedia network.



(a) Results for $K = 3$.

(b) Results for $K = 10$.

Fig. 16: Accuracies of extracting nodes with high anti-majoritarian tendency for the Wikipedia network.

Table 8: Results for estimation errors of anti-majoritarian tendency for the Wikipedia network ($T_0 = 1000$). Note that the two-side 0.05 point of the t-distribution with 9 degrees of freedom is $t^*_{9,0.05} = 2.262$.

	$K = 3$	$K = 3$	$K = 10$	$K = 10$
Method	Average of error $\mathcal{E}_\alpha$	t-value $\mathcal{T}^{PC}_\alpha$	Average of error $\mathcal{E}_\alpha$	t-value $\mathcal{T}^{PC}_\alpha$
proposed	0.0336	—	0.0224	—
uniform value	0.0489	3.2202	0.0360	9.0308
naive	0.1550	51.3607	0.2409	392.8008

network. These results clearly demonstrate the advantage of the proposed method, and it does not seem feasible to detect even roughly the high anti-majoritarian tendency nodes without using the explicit model and solving the optimization problem.

Here, we also note that the proposed method accurately estimated the opinion values. In fact, the average estimation errors of opinion value were less than 1% at $T_0 = 1000$ for all cases. Moreover, we note that the processing times of the proposed method at $T_0 = 1000$ for $K = 3$ and $K = 10$ were less than 3 min. and 4 min., respectively. All our experiments were executed on a single PC with an Intel Core 2 Duo 3GHz processor, with 2GB of memory, running under Linux.

7 Conclusion

Unlike the popular probabilistic model such as Independent Cascade and Linear Threshold models for information diffusion where the node in the network takes only one of the two states (active or inactive), applications such as on-line competitive service in which a user can choose one from multiple choices and opinion formation in which a person listens to his/her neighbors" different opinions and decides whether to change his/her opinion require a model that can handle multiple states.

We extended a voter model, a model of opinion formation dynamics where the basic assumption adopted is that people change their opinions following their neighbors' majority opinion, and proposed a new opinion formation model called Value-weighted Mixture Voter (VwMV) Model to analyze how the multiple opinions spread over a large social network and predict future opinion share. The model has two new features. One is that each opinion can have a value, a measure of opinion's importance, and the other is that each node can have an anti-majoritarian tendency, a measure of deviation from the ordinary behavior. In particular, the latter reflects the fact that there are always people who do not agree with the majority and support the minority opinion. Both are parameters in the model, and their values are not known in general.

Our goal was to 1) learn the parameters from a limited amount of observed opinion propagation data and predict the opinion share in the near future, 2) identify the anti-majoritarians from the learned results, and 3) analyze asymptotic behavior of average opinion dynamics to uncover its intrinsic characteristics.

For the first and the second goals we showed that these parameters are learnable from a sequence of observed opinion data by iteratively maximizing the likelihood function. We further showed that it is enough to learn the opinion values and the average anti-majoritarian tendency in good accuracy if the target is to predict the future opinion share, which can be done easily using a limited amount of observed data, but identifying the anti-majoritarians in good accuracy requires much longer observation data because the anti-majoritarian tendency of each node has to be learned. The learning algorithm is guaranteed to find the global optimal solution when there are no anti-majoritarians but may be trapped to a local optimal solution when there are anti-majoritarians. However, the numerical experiment shows that the algorithm converges to a global optimal if there is enough amount of data. We emphasize that use of the learned model can predict the future opinion share much more accurately than a simple polynomial extrapolation can do, and a model ignoring these parameters (opinion values and the anti-majoritarian tendencies) substantially degrades the performance of share prediction. We tried to find a simpler way to estimate the anti-majoritarian tendency of each node, but there seems to be no way. The heuristic that simply counts the number of opinion updates in which the chosen opinion is the same as the minority opinion gives only a very poor approximation. Thus, it is important to explicitly model the anti-majoritarian tendency to predict the correct future opinion share. For the third goal we applied the mean field theory and uncovered the following features. In a situation where the local opinion share can be approximated by the average opinion share, 1) when there are no anti-majoritarians, the opinion with the highest value eventually takes over, but 2) when there is a certain fraction of anti-majoritarians, it is not necessarily the case that the opinion with the highest value prevails and wins, and further, 3) in both cases, when the opinion values are uniform, the opinion share prediction problem becomes ill-defined and any opinion can win. Although the mean field approximation does not hold in real networks, the simulation that uses the real world network structure supports that this holds for real world social networks that we used in this study. We believe that these findings are useful in deepening our understanding the behavior of opinion dynamics.

References

Agarwal N, Liu H (2008) Blogosphere: Research issues, tools, and applications. SIGKDD Explorations 10:18–31

Arenson AJ (1996) Rejection of the power of judicial review in britain. Deakin Law Review 3:37–53

Bakshy E, Hofman J, Mason W, Watts D (2011) Everyone's an influencer: Quantifying influences on twitter. In: Proceedings of the 4th International Conference on Web Search and Data Mining (WSDM'11), pp 65–74

Castellano C, Munoz MA, Pastor-Satorras R (2009) Nonlinear *q*-voter model. Physical Review E 80:041129:1–041129:8

Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09), pp 199–208

Chen W, Yuan Y, Zhang L (2010) Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of the 10th IEEE International Conference on Data Mining (ICDM'10), pp 88–97

Crandall D, Cosley D, Huttenlocner D, Kleinberg J, Suri S (2008) Feedback effects between similarity and social influence in online communities. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08), pp 160–168

Domingos P (2005) Mining social networks for viral marketing. IEEE Intelligent Systems 20:80–82

Domingos P, Richardson M (2001) Mining the network value of customers. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01), pp 57–66

Donnelly P, Welsh D (1984) The antivoter problem: random 2-colourings of graphs. In: Graph Theory and Combinatorics, pp 133–144

Even-Dar E, Shapira A (2007) A note on maximizing the spread of influence in social networks. In: Proceedings of the 3rd International Workshop on Internet and Network Economics (WINE'07), pp 281–286

Gill J, Gainous J (2002) Why does voting get so complicated? a review of theories for analyzing democratic participation. Statistical Science 17:383–404

Gruhl D, Guha R, Liben-Nowell D, Tomkins A (2004) Information diffusion through blogspace. SIGKDD Explorations 6:43–52

Holme P, Newman MEJ (2006) Nonequilibrium phase transition in the coevolution of networks and opinions. Physical Review E 74:056108:1–056108:5

Huber M, Reinert G (2004) The stationary distribution in the antivoter model: exact sampling and approximations. In: Stein's Method: Expository Lectures and Applications, IMS Lecture Notes, vol 46, pp 75–92

Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03), pp 137–146

Kimura M, Saito K, Motoda H (2009) Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data 3:9:1–9:23

Kimura M, Saito K, Nakano R, Motoda H (2010a) Extracting influential nodes on a social network for information diffusion. Data Mining and Knowledge Discovery 20:70–97

Kimura M, Saito K, Ohara K, Motoda H (2010b) Learning to predict opinion share in social networks. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10), pp 1364–1370

Kimura M, Saito K, Ohara K, Motoda H (2011) Detecting anti-majority opinionists using value-weighted mixture voter model. In: Proceedings of the 14th International Conference on Discovery Science (DS'11), LNAI 6926, pp 150–164

Klimt B, Yang Y (2004) The enron corpus: A new dataset for email classification research. In: Proceedings of the 15th European Conference on Machine Learning (ECML'04), pp 217–226

Leskovec J, Adamic LA, Huberman BA (2007a) The dynamics of viral marketing. ACM Transactions on the Web 1:5:1–5:39

Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N (2007b) Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07), pp 420–429

Liggett TM (1999) Stochastic interacting systems: contact, voter, and exclusion processes. Spriger, New York

Mathioudakis M, Bonch F, Castillo C, Gionis A, Ukkonen A (2011) Sparsification of influence networks. In: Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'11), pp 529–537

Matloff N (1977) Ergodicity conditions for a dissonant voting model. Annals of Probability 5:371–386

Newman MEJ (2003) The structure and function of complex networks. SIAM Review 45:167–256

Newman MEJ, Forrest S, Balthrop J (2002) Email networks and the spread of computer viruses. Physical Review E 66:035101:1–035101:4

Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435:814–818

Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02), pp 61–70

Röllin A (2007) Translated poisson approximation using exchangeable pair couplings. Annals of Applied Probablity 17:1596–1614

Romero D, Meeder B, Kleinberg J (2011) Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In: Proceedings of the 20th International World Wide Web Conference (WWW'11), pp 695–704

Sood V, Redner S (2005) Voter model on heterogeneous graphs. Physical Review Letters 94:178701:1–178701:4

Wu F, Huberman BA (2008) How public opinion forms. In: Proceedings of the 4th International Workshop on Internet and Network Economics (WINE'08), pp 334–341

Yang H, Wu Z, Zhou C, Zhou T, Wang B (2009) Effects of social diversity on the emergence of global consensus in opinion dynamics. Physical Review E 80:046108:1–046108:5

Detecting Changes in Information Diffusion Pattern over Social Network

KAZUMI SAITO, University of Shizuoka
MASAHIRO KIMURA, Ryukoku University
KOUZOU OHARA, Aoyama Gakuin University
HIROSHI MOTODA, Osaka University

We addressed the problem of detecting the change in behavior of information diffusion over a social network which is caused by an unknown external situation change using a small amount of observation data in a retrospective setting. The unknown change is assumed to be effectively reflected in changes in the parameter values in the probabilistic information diffusion model, and the problem is reduced to detecting where in time and how long this change persisted and how big this change is. We solved this problem by searching the change pattern that maximizes the likelihood of generating the observed information diffusion sequences, and in doing so we devised a very efficient general iterative search algorithm using the derivative of the likelihood which avoids parameter value optimization during each search step. This is in contrast to the naive learning algorithm in that it has to iteratively update the patten boundaries, each requiring the parameter value optimization and thus is very inefficient. We tested this algorithm for two instances of the probabilistic information diffusion model which has different characteristics. One is of information push style and the other is of information pull style. We chose asynchronous independent cascade (AsIC) model as the former and value-weighted voter (VwV) model as the latter. The AsIC is the model for general information diffusion with binary states and the parameter to detect its change is diffusion probability and the VwV is the model for opinion formation with multiple states and the parameter to detect its change is opinion value. The results tested on these two models using four real world network structures confirmed that the algorithm is robust enough and can efficiently identify the correct change pattern of the parameter values. Comparison with the naive method that finds the best combination of change boundaries by an exhaustive search through a set of randomly selected boundary candidates showed that the proposed algorithm far outperforms the native method both in terms of accuracy and computation time.

Categories and Subject Descriptors: G.2.2 [**Discrete Mathematics**]: Graph Theory—*Network problems*; H.2.8 [**Database Management**]: Database Applications—*Data mining*; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*sociology*

General Terms: Algorithms

Additional Key Words and Phrases: Change point detection, information diffusion, parameter learning, social networks

ACM Reference Format:
Saito, K., Kimura, M., Ohara, K., and Motoda, H. 2011. Detecting changes in information diffusion pattern over social network. ACM Trans. Intell. Syst. Technol. V, N, Article A (January YYYY), 23 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-11-4111, and JSPS Grant-in-Aid for Scientific Research (C) (No. 23500194). Author's addresses: K. Saito, School of Administration and Informatics, University of Shizuoka; M. Kimura, Department of Electronics and Informatics, Ryukoku University; K. Ohara, Department of Integrated Information Technology, Aoyama Gakuin University; H. Motoda, Institute of Scientific and Industrial Research, Osaka University.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Recent technological innovation in the web such as blogosphere and knowledge/media-sharing sites is remarkable, which has made it possible to form various kinds of large social networks, through which behaviors, ideas, rumors and opinions can spread, and our behavioral patterns are to a considerable degree affected by the interaction with these networks and substantial attention has been directed to investigating the spread of information in these networks [Newman et al. 2002; Newman 2003; Gruhl et al. 2004; Domingos 2005; Leskovec et al. 2006; Crandall et al. 2008; Wu and Huberman 2008].

These studies have shown that it is important to consider the diffusion mechanism explicitly and the measures based on network structure alone, *i.e.*, various centrality measure, are not enough to identify the important nodes [Kimura 2009; 2010a]. Information diffusion is modeled typically by probabilistic models. Most representative and fundamental ones for general information diffusion are independent cascade (IC) model [Goldenberg et al. 2001; Kempe et al. 2003], linear threshold (LT) model [Watts 2002; Watts and Dodds 2007] and their extensions that include incorporating asynchronous time delay [Saito et al. 2009b; 2010a]. The IC model is a model of information push style, *i.e.*, the information sender (a node) tries to push the information to the neighboring receivers (child nodes) in a probabilistic way. The LT model is a model of information pull style, *i.e.*, the information receiver (a node) tries to pull the information from the neighboring senders (parents nodes) in a probabilistic way. Since the focus of study is "influence", these models assume binary states, *i.e.*, nodes are either active (influenced) or inactive (uninfluenced). Explicit use of these models to solve such problems as the *influence maximization problem* [Kempe et al. 2003; Kimura et al. 2010a; Chen et al. 2010a; 2010b] and the *contamination minimization problem* [Kimura et al. 2009] clearly shows the advantage of the model. The identified influential nodes and links are considerably different from the ones identified by the centrality measures. Another type of information diffusion model that is also often used is voter model [Even-Dar and Shapria 2007] and its extensions that include incorporating opinion values [Kimura et al. 2010b], node strength [Yamagishi et al. 2011] and anti-majoritarian tendency [Kimura et al. 2011]. The voter model is a model of information pull style and is used to study the spread of opinions, *i.e.*, opinion formation. It is similar to the LT model in that the opinion of a person is affected by the opinions of his/her neighbors. What is different from the LT model is that it has to have multiple states if it has to deal with multiple opinions[1]. This notion is not necessarily limited to opinion. Application such as an on-line competitive service in which a user can choose one from multiple choices/decisions requires a model that handles multiple states. There has been a variety of work on the voter model, too. Dynamical properties of the basic model have been extensively studied including how the degree distribution and the network size affect the mean time to reach consensus from mathematical point of view [Liggett 1999; Sood and Redner 2005]. Several variants of the voter model are also investigated and non equilibrium phase transition is analyzed [Castellano et al. 2009; Yang et al. 2009] from physics point of view. Yet another line of work extends the voter model by combining it with a network evolution model [Holme and Newman 2006; Crandall et al. 2008]. Kimura et al. [2010b] analyzed how the opinion values affect the opinion share dynamics in their recent study.

What is common to all the above models is that they are all probabilistic models and have parameters to characterize the information diffusion. The parameters must be known in advance for the model to be usable for analysis. It is generally difficult

[1] The basic voter model has only two opinions but it is straightforward to extend it to handle multiple opinions.

to determine the values of these parameters theoretically, and thus, attempts have been made to learn these parameter values by observing the information diffusion sequence data [Saito et al. 2009a; 2009b; 2010a; 2010b; Gomez-Rodriguez et al. 2010; Myers and Leskovec 2010; Kimura et al. 2010b]. In essence the likelihood of generating the observed data by the model employed is first derived, and then the parameter values are determined such that the likelihood is maximized. In particular, Myers and Leskovec [2010] showed that for a certain class of diffusion models, the problem can effectively be transformed to a convex programming for which a global solution is guaranteed. Another important common assumption made in these studies is that the model is stationary. Since the model is probabilistic, even if the model is stationary, the way information propagates from a particular node is not the same (not deterministic) and each time the diffusion result is different. However, the model parameter values remain the same during the whole course of analysis.

This paper addresses a different aspect of information diffusion, and extends and integrates our recent studies [Saito et al. 2011a; Ohara et al. 2011]. We note that our behavior is affected not only by the behavior of our neighbors but also by other external factors. The model only accounts for the interaction with neighbors. The behavior we observe includes both effects. The problem we address here is to detect the change in the model from a limited amount of observed information diffusion data. If this is possible, this would bring a substantial advantage. For example, we can infer that something unusual happened during a particular period of time by simply analyzing the limited amount of data.

This is in some sense the same, in the spirit, with the work by Kleinberg [2002] and Swan and Allan [2000]. They noted a huge volume of the data stream, tried to organize it and extract structures behind it. This is done in a retrospective framework, *i.e.*, assuming that there is a flood of abundant data already and there is a strong need to understand it. Kleinberg's work is motivated by the fact that the appearance of a topic in a document stream is signaled by a "burst of activity" and identifying its nested structure manifests itself as summarization of the activities over a period of time, making it possible to analyze the underlying content much easier. He used a hidden Markov model in which bursts appear naturally as state transitions, and successfully identified the hierarchical structure of e-mail messages. Swan and Allan's work is motivated by the need to organize huge amount of information in an efficient way. They used a statistical model of feature occurrence over time based on hypotheses testing and successfully generated clusters of named entities and noun phrases that capture the information corresponding to major topics in the corpus, and designed a way to nicely display the summary on the screen (Overview Timelines). Our aim is not exactly the same as theirs. We are interested in detecting changes in the external factors which are hidden/embedded in the data. We also follow the same retrospective approach, *i.e.*, we are not predicting the future, but we are trying to understand the phenomena that happened in the past. There are many factors that bring in changes and evidently the model cannot accommodate all of them. We formalize this as the unknown changes in the parameter value of the diffusion model we employ, and we reduce the problem to that of detecting where in time and how long this change persisted and how big this change is. We call the period where the parameter takes anomalous values as "hot span" and the rest as "normal span".

We have chosen the asynchronous independent cascade (AsIC) model [Saito et al. 2009b; 2010a] as the one that represents the model of information push style, and the value-weighted voter (VwV) model [Kimura et al. 2010b] as the one that represents the model of information pull style[2]. As explained above, the AsIC is the model for general information diffusion with binary states and the parameter to detect its change is diffusion probability and the VwV is the model for opinion formation with multiple states and the parameter to detect its change is opinion value. These two models are recalled in Section 2. We generalized the parameter optimization algorithm that was first introduced in [Saito et al. 2011a; Ohara et al. 2011] so that it can cover both the models as two different instances and expanded the experiments to verify that the same algorithm works satisfactorily for two different types of information diffusion models. As in our previous work, we limit the form of change to a rect-linear one, that is, the parameter value changes to a new large value, persists for a certain period of time and is restored to the original value and stays the same thereafter [3]. In this simplified setting, detecting the hot span is equivalent to identifying the time window where the parameter value is anomalous and estimating the parameter values both in the hot and the normal spans.

We use the same parameter optimization algorithm as in [Saito et al. 2009b; Kimura et al. 2010b], *i.e.*, the EM-like algorithm for the AsIC model that iteratively updates the values to maximize the model's likelihood of generating the observed data sequences, and the Newton method for the VwV model that guarantees globally maximizing the likelihood. However, the problem here is more difficult because it has another loop to search for the hot span on top of the above loop. The naive learning algorithm has to iteratively update the patten boundaries (outer loop) and the value must also be optimized for each combination of the pattern boundaries (inner loop), which is extraordinary inefficient. Our main contribution is that we devised a very efficient general search algorithm which works for probabilistic information diffusion models and avoids the inner loop optimization by using the information of the first order derivative of the likelihood with respect to the parameters. We tested its performance using the structures of four real world networks (Blog, Coauthorship, Enron and Wikipedia), and confirmed that the algorithm can efficiently identify the hot span correctly as well as the parameter values of both the normal and the hot spans. We further compared our algorithm with the naive method that finds the best combination of the hot span boundaries by an exhaustive search from a set of randomly selected boundary candidates, and showed that the proposed algorithm far outperforms the naive method both in terms of accuracy and computation time.

The paper is organized as follows. After very briefly introducing the two diffusion models, AsIC and VwV in Section 2, we define the problem in Section 3 and recall how the parameters can be learned in each model in Section 4. The main part is Section 5 where we explain how we efficiently search for the hot span as well as the parameter values. The results are explained in Section 6, followed by discussion in Section 7. We end this paper by summarizing the main result in Section 8.

2. INFORMATION DIFFUSION MODELS

We focus on two types of information diffusion model on a social network $G = (V, E)$, where V and E ($\subset V \times V$) are the sets of all the nodes and the links, respectively. One is the asynchronous independent cascade (AsIC) model that is an extension of the independent cascade (IC) model, and the other is the value-weighted voter (VwV) model that is an extension of the standard voter model. They were extended to meet more realistic situations. We recall their definitions below.

[2] We could have chosen AsLT instead of VwV. There is no specific reason that we cannot handle AsLT. Our aim is to show that our approach is general enough and applicable to a wide variety of diffusion models.

[3] We discuss that the basic algorithm can be extended to more general change patterns in Section 7, and show that it works for two distinct rect-linear patterns in case of AsIC.

2.1. Asynchronous Independent Cascade (AsIC) Model

The AsIC model we use in this paper incorporates asynchronous time delay into the IC model which does not account for time-delay, noting that each node changes its state asynchronously in reality [Saito et al. 2009b; 2010a]. Here, we consider choosing a delay-time from the exponential distribution for the sake of convenience, but of course other distributions such as power-law and Weibull can be employed.

For the AsIC model, the underlying network $G = (V, E)$ is a directed graph. For any $v \in V$, the set of all the nodes that have links from v (child nodes) is denoted by

$$F(v) = \{u \in V;\ (v, u) \in E\},$$

and the set of all the nodes that have links to v (parent nodes) is denoted by

$$B(v) = \{u \in V;\ (u, v) \in E\}.$$

Each node has one of the two states (active and inactive), and the nodes are called *active* if they have been influenced. It is assumed that nodes can switch their states only from inactive to active.

The AsIC model has two types of parameters $p_{u,v}$ and $r_{u,v}$ with $0 < p_{u,v} < 1$ and $r_{u,v} > 0$, where $p_{u,v}$ and $r_{u,v}$ are referred to as the *diffusion probability* through link (u, v) and the *time-delay parameter* through link (u, v), respectively. We define the *diffusion-probability vector* $\boldsymbol{p}$ and the *time-delay parameter vector* $\boldsymbol{r}$ by

$$\boldsymbol{p} = (p_{u,v})_{(u,v)\in E}, \quad \boldsymbol{r} = (r_{u,v})_{(u,v)\in E}.$$

The information diffusion process unfolds in continuous-time t, and proceeds from a given initial active node in the following way. When a node u becomes active at time t, it is given a single chance to activate each currently inactive node $v \in F(u)$. A delay-time δ is chosen from the exponential distribution with parameter $r_{u,v}$. The node u attempts to activate the node v if v has not been activated by time $t + \delta$, and succeeds with probability $p_{u,v}$. If u succeeds, v will become active at time $t + \delta$. The information diffusion process terminates if no more activations are possible.

2.2. Value weighted Voter (VwV) Model

The mathematical model we use for the diffusion of opinions is the VwV model with K (≥ 2) opinions [Kimura et al. 2010b]. For the VwV model, the underlying network $G = (V, E)$ is an undirected (bidirectional) graph with self-loops. For a node $v \in V$, let $\Gamma(v)$ denote the set of neighbors of v in G, that is,

$$\Gamma(v) = \{u \in V;\ (u, v) \in E\}.$$

Note that $v \in \Gamma(v)$ because of the existence of self-loops.

In the VwV model, each node of G is endowed with $(K + 1)$ states; opinions $1, \cdots,$ K, and *neutral* (*i.e.*, no-opinion state). It is assumed that a node never switches its state from any opinion k back to neutral. The model has a parameter w_k (> 0) for each opinion k, which is called the *opinion value* and must be estimated from observed opinion diffusion data. We define the *opinion-value vector* $\boldsymbol{w}$ by

$$\boldsymbol{w} = (w_1, \cdots, w_K).$$

Let $f_t : V \rightarrow \{0, 1, 2, \cdots, K\}$ denote the opinion distribution at time t, where $f_t(v)$ stands for the opinion of node v at time t, and opinion 0 denotes the neutral state. We also denote by $n_k(t, v)$ the number of v's neighbors that hold opinion k as the latest one before time t for $k = 1, 2, \cdots, K$, *i.e.*,

$$n_k(t, v) = |\{u \in \Gamma(v);\ \phi_t(u) = k\}|,$$

where $\phi_t(u)$ is the latest opinion of u before time t.

Given a target time T, and an initial state in which each opinion is assigned to only one distinct node and all other nodes are in the neutral state, the evolution process of the model unfolds in the following way. At time 0, each node v independently decides its update time t according to some probability distribution such as an exponential distribution with parameter r_v, where r_v becomes also a model parameter and then we define the time-delay parameter vector $\boldsymbol{r}$ by $\boldsymbol{r} = (r_v)_{v \in V}$. The successive update time is determined similarly at each update time t. Node v changes its opinion at its update time t as follows: If node v has at least one neighbor with some opinion before time t, $f_t(v) = k$ with probability $w_k n_k(t, v) \;/\; \sum_{k'=1}^{K} w_{k'}\, n_{k'}(t, v)$ for $k = 1, \cdots, K$, otherwise, $f_t(v) = 0$ with probability 1. It is noted that since node v is included in its neighbors by definition, its own opinion is also reflected. The process is repeated from the initial time $t = 0$ until the next update-time attains a given final-time T.

3. PROBLEM DEFINITION

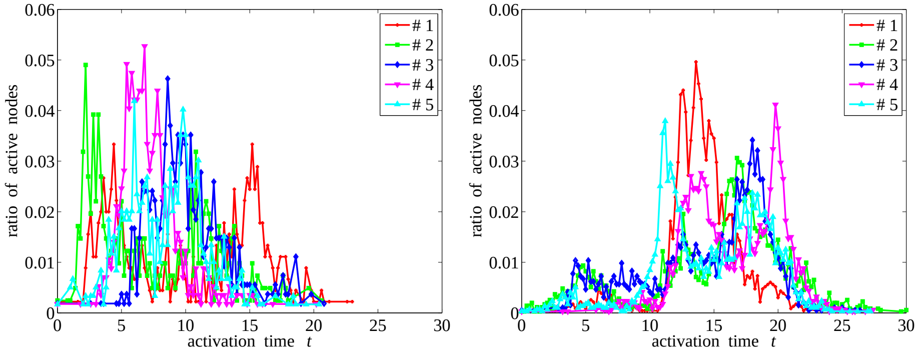
We address the *hot span detection problem*. In this problem, we assume that some change has happened in the way the information diffuses, and we observe the diffusion sequences of a certain topic in which the change is embedded, and consider detecting where in time and how long this change persisted and how big this change is. In the following subsections, we describe a specific detection problem by focusing on the above diffusion models, *i.e.*, the AsIC model and the VwV model.

3.1. AsIC Model

An information diffusion result generated by the AsIC model is represented as a set of pairs of active nodes and their activation times; *i.e.*, $\{(u, t_u), (v, t_v), \cdots\}$. We consider a diffusion result $\mathcal{D}(0, T)$, where the initial activation time is set to 0 and the final observation time is denoted by T. Since we employ only a single diffusion result $\mathcal{D}(0, T)$, we place a constraint that $p_{u,v}$ and $r_{u,v}$ do not depend on link (u, v), *i.e.*, $p_{u,v} = p$, $r_{u,v} = r$ ($\forall (u, v) \in E$), which should be acceptable noting that we can naturally assume that people behave quite similarly when talking about the same topic (see Section 7).

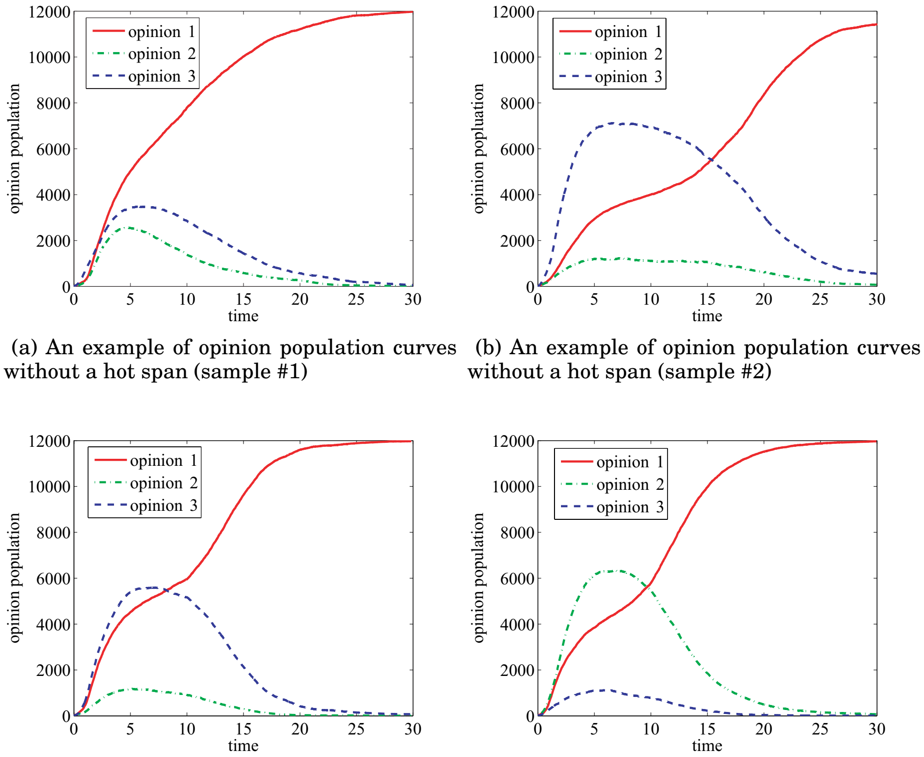
Let $[T_1, T_2)$ denote the hot span of the information diffusion, and let p_n and p_h denote the diffusion probability for the normal span and the hot span, respectively. Namely, the diffusion probability p is obtained by $p = p_n$ for the period $[0, T_1)$, $p = p_h$ for the period $[T_1, T_2)$, and $p = p_n$ for the period $[T_2, T)$. Here we assume for simplicity that the time-delay parameter r does not change and takes the same value for the entire period $[0, T)$. Then, the hot span detection problem is reduced to detecting the hot span $[T_1, T_2)$ and estimating p_n and p_h from the observed diffusion result $\mathcal{D}(0, T)$.

Figure 1 shows five examples of diffusion sample with (Fig. 1b) and without (Fig. 1a) a hot span based on the AsIC model, where the parameters are set at $p_n = 0.1$, $p_h = 0.3$, $r = 1.0$, $T_1 = 10$, $T_2 = 20$. The network used is the blog network described later in Subsection 6.1. We plotted the ratio of active nodes (the number of nodes activated at time step t divided by the number of total active nodes over the whole time span) for five independent simulations, each from a randomly chosen initial source node at time $t = 0$. Comparing these two figures, we can clearly see bursty activities around the hot span $[10, 20)$ in Fig. 1b. However, each curve in Fig. 1b behaves differently, *i.e.*, some has its bursty activities only in the first half, some other has them only in the last half, and yet some other has two peaks during the hot span. This means that it is quite difficult to accurately detect the true hot span from only a single diffusion sample. Methods that use only the observed bursty activities, including those proposed by Swan and Allan [2000] and Kleinberg [2002] would not work.



(a) Diffusion samples without a hot span (b) Diffusion samples with a hot span

Fig. 1: Information diffusion in the blog network for the AsIC model. Results of five independent runs are shown.



(a) An example of opinion population curves without a hot span (sample #1)

(b) An example of opinion population curves without a hot span (sample #2)

(c) An example of opinion population curves with a hot span (sample #1)

(d) An example of opinion population curves with a hot span (sample #2)

Fig. 2: Information diffusion in the blog network for the VwV model.

3.2. VwV Model

Similarly to the detection problem for the AsIC model, let $[T_1, T_2)$ denote the hot span of the diffusion of opinions under the VwV model. Recall that this implies that the intervals $[0, T_1)$ and $[T_2, T)$ are the normal spans. We place the same assumption that there is no change in the value of the time-delay parameter vector $\boldsymbol{r}$ for simplicity. Let $\boldsymbol{w}_n$ and $\boldsymbol{w}_h$ denote the opinion-value vectors for the normal span and the hot span, respectively. Note that $\boldsymbol{w}_n/||\boldsymbol{w}_n|| \neq \boldsymbol{w}_h/||\boldsymbol{w}_h||$ since the opinion dynamics under the VwV model is invariant to positive scaling of the opinion-value vector $\boldsymbol{w}$, where $\|\boldsymbol{w}_n\|$ and $\|\boldsymbol{w}_h\|$ stand for the norm of vectors $\boldsymbol{w}_n$ and $\boldsymbol{w}_h$, respectively. Then, the change detection problem is formulated as follows: Given the opinion diffusion data $\mathcal{D}(0, T)$ in time-interval $[0, T)$, detect the hot span $[T_1, T_2)$, and estimate the opinion-value vector $\boldsymbol{w}_h$ of the hot span and the opinion-value vector $\boldsymbol{w}_n$ of the normal span. Here, $\mathcal{D}(0, T)$ consists of a sequence of (v, t, k) such that node v changed its opinion to opinion k at time t.

Figure 2 shows two examples of opinion diffusion sample with (Figs. 2c and 2d) and without (Figs. 2a and 2b) a hot span based on the VwV model with $K = 3$ opinions, where the opinion-value vectors are set at $\boldsymbol{w} = (2.0, 1.0, 1.0)$ for Figs. 2a and 2b, and $\boldsymbol{w}_n = (2.0, 1.0, 1.0)$, $\boldsymbol{w}_h = (3.0, 1.0, 1.0)$, $T_1 = 10$ and $T_2 = 20$ for Figs. 2c and 2d. The network used is the same blog network as in Fig. 1. We plotted the population of each opinion k, $|\{v \in V; f_t(v) = k\}|$, as a function of time t. It must be difficult to know the existence of a hot span from only their curves depicted in Figs. 2b and 2d. Moreover, since the VwV model is a stochastic process model, every sample of opinion diffusion can behave differently. Again, this means that it is quite difficult to accurately detect the true hot span from only a single sample of opinion diffusion. We believe that an explicit use of underlying opinion diffusion model is essential to solve this problem. It is crucially important to detect the hot span precisely in order to identify the external factors which caused the behavioral changes.

4. MODEL PARAMETER LEARNING

We describe the framework of model parameter learning as a likelihood maximization problem for the AsIC and the VwV models.

4.1. Parameter Learning for AsIC Model

First, we consider estimating the values of diffusion probability p and time-delay parameter r from an observed diffusion result $\mathcal{D}(0, T) = \{\cdots, (v, t_v), \cdots\}$ when there is no hot span. Recall that the initial activation time is set to 0 and the final observation time is denoted by T. Let $\mathcal{D}$ be the set of all the activated nodes in $\mathcal{D}(0, T)$, *i.e.*,

$$\mathcal{D} = \{v \in V;\ (v, t_v) \in \mathcal{D}(0, T)\}.$$

For each node $v \in \mathcal{D}$, let $\mathcal{A}_v$ be the set of its parent nodes that had a chance to activate it, *i.e.*,

$$\mathcal{A}_v = \{u \in B(v);\ (u, t_u) \in \mathcal{D}(0, T), t_u < t_v\}.$$

Although we place a constraint that $p_{u,v} = p$, $r_{u,v} = r$ $(\forall (u, v) \in E)$, we develop a general theory in terms of $\boldsymbol{p}$ and $\boldsymbol{r}$ to be consistent with the description in Subsection 5.2. Let $\mathcal{X}_{u,v}(p_{u,v}, r_{u,v})$ denote the probability density that a node $u \in \mathcal{A}_v$ activates the node v at time t_v, that is,

$$\mathcal{X}_{u,v}(p_{u,v}, r_{u,v}) \ = \ p_{u,v}\, r_{u,v} \exp(-r_{u,v}(t_v - t_u)). \tag{1}$$

Let $\mathcal{Y}_{u,v}(p_{u,v}, r_{u,v})$ denote the probability that the node v is not activated by a node $u \in \mathcal{A}_v$ within the time-period (t_u, t_v), that is,

$$\begin{aligned} \mathcal{Y}_{u,v}(p_{u,v}, r_{u,v}) \ &= \ 1 - p_{u,v} \int_{t_u}^{t_v} r_{u,v} \exp(-r_{u,v}(t - t_u)) dt \\ &= \ p_{u,v} \exp(-r_{u,v}(t_v - t_u)) + (1 - p_{u,v}). \end{aligned} \tag{2}$$

By using Eqs. (1) and (2), we can obtain the probability density $h_v(\boldsymbol{p}, \boldsymbol{r})$ that a node v is activated at time t_v,

$$h_v(\boldsymbol{p}, \boldsymbol{r}) \;=\; \sum_{u \in \mathcal{A}_v} \mathcal{X}_{u,v}(p_{u,v}, r_{u,v}) \left(\prod_{z \in \mathcal{A}_v \setminus \{u\}} \mathcal{Y}_{z,v}(p_{z,v}, r_{z,v}) \right), \tag{3}$$

and the probability $\psi_{v,z}(p_{v,z}, r_{v,z})$ that a node z is not activated by a node v within $[0, T)$,

$$\psi_{v,z}(p_{v,z}, r_{v,z}) \;=\; p_{v,z} \exp(-r_{v,z}(T - t_v)) + (1 - p_{v,z}). \tag{4}$$

Then, from Eqs. (3) and (4), the following log likelihood function $\mathcal{L}(\boldsymbol{p}, \boldsymbol{r}; \mathcal{D}(0,T))$ can be obtained for observed data $\mathcal{D}(0,T)$

$$\mathcal{L}(\boldsymbol{p}, \boldsymbol{r}; \mathcal{D}(0,T)) \;=\; \sum_{v \in \mathcal{D}} \left(\log h_v(\boldsymbol{p}, \boldsymbol{r}) + \sum_{z \in F(v) \setminus \mathcal{D}} \log \psi_{v,z}(p_{v,z}, r_{v,z}) \right). \tag{5}$$

Here, we recall $p_{u,v} = p$, $r_{u,v} = r$ for any $(u, v) \in E$. The values of parameters p and r can be stably obtained by maximizing Eq. (5) using an EM-like algorithm (see Appendix A for more details).

Now, we assume that there exists a hot span $S = [T_1, T_2)$. Let $p(t)$ denote the value of parameter p at time t. According to our problem setting, we consider the parameter switching,

$$p(t) \;=\; \begin{cases} p_n & \text{if } \; t \in [0, T) \setminus S, \\ p_h & \text{if } \; t \in S. \end{cases}$$

For the hot span S, we split the set of the active nodes $\mathcal{D}$ as follows:

$$\mathcal{D}_n(S) \;=\; \{v \in \mathcal{D};\; t_v \in [0,T) \setminus S\},$$
$$\mathcal{D}_h(S) \;=\; \{v \in \mathcal{D};\; t_v \in S\}.$$

For any $v \in \mathcal{D}$, let $h_v(p_n, p_h, r; S)$ be the probability density that node v is activated at time t_v when there exists hot span S. By using Eqs. (1) and (2), we obtain

$$\begin{aligned} & h_v(p_n, p_h, r; S) \\ &= \sum_{u \in \mathcal{A}_v \cap \mathcal{D}_n(S)} \mathcal{X}_{u,v}(p_n, r) \left(\prod_{z \in \mathcal{A}_v \cap \mathcal{D}_n(S) \setminus \{u\}} \mathcal{Y}_{z,v}(p_n, r) \prod_{z \in \mathcal{A}_v \cap \mathcal{D}_h(S)} \mathcal{Y}_{z,v}(p_h, r) \right) \\ &+ \sum_{u \in \mathcal{A}_v \cap \mathcal{D}_h(S)} \mathcal{X}_{u,v}(p_h, r) \left(\prod_{z \in \mathcal{A}_v \cap \mathcal{D}_n(S)} \mathcal{Y}_{z,v}(p_n, r) \prod_{z \in \mathcal{A}_v \cap \mathcal{D}_h(S) \setminus \{u\}} \mathcal{Y}_{z,v}(p_h, r) \right). \end{aligned} \tag{6}$$

Using Eqs. (4) and (6), we can define an objective function $\mathcal{L}(p_n, p_h, r; \mathcal{D}(0,T), S)$ for the hot span detection problem by adequately modifying Eq. (5) under the switching scheme as follows:

$$\begin{aligned} & \mathcal{L}(p_n, p_h, r; \mathcal{D}(0,T), S) \\ & \quad = \sum_{v \in \mathcal{D}} \log h_v(p_n, p_h, r; S) + \sum_{v \in \mathcal{D}_n(S)} \sum_{z \in F(v) \setminus \mathcal{D}} \log \psi_{v,z}(p_n, r) \\ & \qquad + \sum_{v \in \mathcal{D}_h(S)} \sum_{z \in F(v) \setminus \mathcal{D}} \log \psi_{v,z}(p_h, r). \end{aligned} \tag{7}$$

Clearly, $\mathcal{L}(p_n, p_h, r; \mathcal{D}(0,T), S)$ is expected to be maximized by setting S to the true hot span $S^* = [T_1^*, T_2^*)$ if a substantial amount of data $\mathcal{D}(0,T)$ is available. Thus, our hot span detection problem is formalized as the following maximization problem:

$$\hat{S} \; = \; \arg\max_{S} \mathcal{L}(\hat{p}_n(S), \hat{p}_h(S), \hat{r}(S); \mathcal{D}(0,T), S), \tag{8}$$

where $\hat{p}_n(S)$, $\hat{p}_h(S)$, and $\hat{r}(S)$ denote the maximum likelihood estimators for a given S.

4.2. Parameter Learning for VwV Model

We also consider estimating the value of opinion-value vector $\boldsymbol{w}$ from an observed opinion diffusion data $\mathcal{D}(0,T)$ in time interval $[0,T)$ (a single example) when there is no hot span[4]. From the evolution process of the model, we can obtain the following log likelihood function

$$\mathcal{L}(\boldsymbol{w}; \mathcal{D}(0,T)) = \log \prod_{(v,t,k)\in\mathcal{C}(0,T)} \frac{n_k(t,v)w_k}{\sum_{k'=1}^{K} n_{k'}(t,v)w_{k'}}, \tag{9}$$

where

$$\mathcal{C}(0,T) = \{(v,t,f_t(v)) \in \mathcal{D}(0,T);\ |\{u \in \Gamma(v);\ f_t(u) \neq 0\}| \geq 2\}.^{5}$$

Thus, our estimation problem is formulated as a maximization problem of the log likelihood function $\mathcal{L}(\boldsymbol{w}; \mathcal{D}(0,T))$ with respect to $\boldsymbol{w}$. We find the optimal value of $\boldsymbol{w}$ by employing a standard Newton method (see Appendix B for more details).

Now, we assume that there exists a hot span $S = [T_1, T_2)$. Let $\boldsymbol{w}(t)$ denote the value of opinion-value vector $\boldsymbol{w}$ at time t. We also consider the following parameter vector switching:

$$\boldsymbol{w}(t) \; = \; \begin{cases} \boldsymbol{w}_n & \text{if } \ t \in [0,T) \setminus S, \\ \boldsymbol{w}_h & \text{if } \ t \in S. \end{cases}$$

For $\forall T_s, T_e$ with $0 \leq T_s < T_e \leq T$, we denote by $\mathcal{D}(T_s, T_e)$ the opinion diffusion data in time interval $[T_s, T_e)$; *i.e.*,

$$\mathcal{D}(T_s, T_e) = \{(v,t,k) \in \mathcal{D}(0,T);\ t \in [T_s, T_e)\}. \tag{10}$$

Then, similarly to the case of the AsIC model, an objective function $\mathcal{L}(\boldsymbol{w}_n, \boldsymbol{w}_h; \mathcal{D}(0,T), S)$ can be defined for the hot span detection problem by adequately modifying Eq. (9) under this switching scheme as follows:

$$\mathcal{L}(\boldsymbol{w}_n, \boldsymbol{w}_h; \mathcal{D}(0,T), S) \; = \; \mathcal{L}(\boldsymbol{w}_n; \mathcal{D}(0,T_1) \cup \mathcal{D}(T_2,T)) + \mathcal{L}(\boldsymbol{w}_h; \mathcal{D}(T_1,T_2)). \tag{11}$$

Again, the extended objective function is expected to be maximized by setting S to be the true span $S^* = [T_1^*, T_2^*)$, provided that $\mathcal{D}(0,T)$ is generated by the VwV model with hot span S^* and is sufficiently large. Therefore, our hot span detection problem is formalized as the following maximization problem:

$$\hat{S} \; = \; \arg\max_{S} \mathcal{L}(\hat{\boldsymbol{w}}_n(S), \hat{\boldsymbol{w}}_h(S); \mathcal{D}(0,T), S), \tag{12}$$

where $\hat{\boldsymbol{w}}_n(S)$ and $\hat{\boldsymbol{w}}_h(S)$ denote the maximum likelihood estimators for a given S.

[4] The time-delay parameter vector $\boldsymbol{r}$ can simply be estimated by averaging the time intervals for each node, and thus excluded from the estimation problem.

[5] We use only those observed data in which there is at least one neighbor that has an opinion.

5. CHANGE DETECTION METHODS

We propose a general method of detecting a hot span that is applicable to both the AsIC model and the VwV model. In order to obtain the optimal hot span $\hat{S}$ according to either Eq. (8) or Eq. (12), we need to prepare a reasonable set of candidate hot spans, denoted by $\mathcal{H}$. One way of doing so is to construct $\mathcal{H}$ by considering all pairs of observed activation (or opinion change) time points. In general, let $\mathcal{T}$ denote the set of all the observed activation (or opinion change) time points,

$$\mathcal{T} = \{t_0, t_1, \cdots, t_N\}, \quad (0 = t_0 < t_1 < \cdots < t_N < T).$$

Then, we can construct a set of candidate hot spans by

$$\mathcal{H} \;=\; \{S = [T_1, T_2);\; T_1 < T_2,\; T_1 \in \mathcal{T},\; T_2 \in \mathcal{T}\}.$$

Hereafter, we denote the model parameter vector by $\boldsymbol{\theta}$; *i.e.*, $\boldsymbol{\theta} = (p, r)$ for the AsIC model and $\boldsymbol{\theta} = \boldsymbol{w}$ for the VwV model. Since the parameter vector $\boldsymbol{\theta}$ is a function of time t in our problem setting, we denote by $\boldsymbol{\theta}(t)$ the value of $\boldsymbol{\theta}$ at time t. Given a hot span $S = [T_1, T_2)$, we consider the following parameter vector switching:

$$\boldsymbol{\theta}(t) \;=\; \begin{cases} \boldsymbol{\theta}_n & \text{if } \; t \in [0, T) \setminus S, \\ \boldsymbol{\theta}_h & \text{if } \; t \in S. \end{cases}$$

Let $S^* = [T_1^*, T_2^*)$ be the true hot span. We assume that observed data $\mathcal{D}(0, T)$ is generated by using the parameter vector $\boldsymbol{\theta}^*(t)$ of hot span S^*. In what follows, after introducing a naive method, we describe our proposed detection method.

5.1. Naive Method

Both Eq. (8) and Eq. (12) can be solved by a naive method which has two iterative loops. In the inner loop we first obtain the maximum likelihood estimators, $\hat{\boldsymbol{\theta}}_n$ and $\hat{\boldsymbol{\theta}}_h$, for each candidate S by maximizing the objective function $\mathcal{L}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_h; \mathcal{D}(0, T), S)$ using either the EM-like algorithm or the Newton method. In the outer loop we select the optimal $\hat{S}$ which gives the largest $\mathcal{L}(\hat{\boldsymbol{\theta}}_n, \hat{\boldsymbol{\theta}}_h; \mathcal{D}(0, T), S)$ value. However, this method can be extremely inefficient when the number of candidate spans is large. Thus, in order to make it work with a reasonable computational cost, we consider restricting the number of candidate time points to a smaller value, denoted by J, *i.e.*, we construct $\mathcal{T}_J$ $(\subset \mathcal{T})$ by selecting J points from $\mathcal{T}$; then we construct a restricted set of candidate spans by

$$\mathcal{H}_J \;=\; \{S = [T_1, T_2);\; T_1 < T_2, T_1 \in \mathcal{T}_J, T_2 \in \mathcal{T}_J\}.$$

Note that $|\mathcal{H}_J| = J(J-1)/2$, which is large when J is large.

5.2. Proposed Method

It is easily conceivable that the naive method can detect the hot span with a reasonably good accuracy when we set J large at the expense of the computational cost, but the accuracy becomes poorer when we set J smaller to reduce the computational load. We propose a novel detection method below which alleviates this problem and can efficiently and stably detect a hot span from the diffusion result $\mathcal{D}(0, T)$.

We first obtain the maximum likelihood estimators, $\hat{\boldsymbol{\theta}}$, based on the original objective function of either Eq. (5) or Eq. (9). Next, we focus on the first-order derivative of the objective function $\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}(0, T))$ with respect to the parameter vector $\boldsymbol{\theta}$ in each observation interval $[t_{j-1}, t_j)$. More specifically, we define a function $\tilde{\mathcal{L}}(\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_N; \mathcal{D}(0, T))$ of $\boldsymbol{\theta}_1$, $\cdots$, $\boldsymbol{\theta}_N$ by

$$\tilde{\mathcal{L}}(\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_N; \mathcal{D}(0, T)) = \mathcal{L}(\tilde{\boldsymbol{\theta}}(t); \mathcal{D}(0, T)),$$

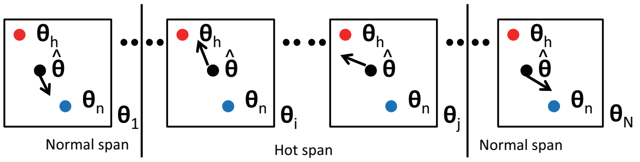


Fig. 3: Direction of the gradient vector at $\hat{\boldsymbol{\theta}}$ in the normal and the hot span.

where $\tilde{\boldsymbol{\theta}}(t) = \boldsymbol{\theta}_j$ if $t \in [t_{j-1}, t_j)$, $(j = 1, \cdots, N)$. Since $\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}(0,T)) = \tilde{\mathcal{L}}(\boldsymbol{\theta}, \cdots, \boldsymbol{\theta}; \mathcal{D}(0,T))$ and $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimator based on $\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}(0,T))$, *i.e.*, when no change in $\boldsymbol{\theta}$ is assumed, we have

$$0 = \frac{\partial \mathcal{L}(\hat{\boldsymbol{\theta}}; \mathcal{D}(0,T))}{\partial \boldsymbol{\theta}} = \sum_{j=1}^{N} \frac{\partial \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}, \cdots, \hat{\boldsymbol{\theta}}; \mathcal{D}(0,T))}{\partial \boldsymbol{\theta}_j} \qquad (13)$$

Note that $\tilde{\mathcal{L}}(\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_N; \mathcal{D}(0,T))$ can be expected to attain the maximum when each $\boldsymbol{\theta}_j$ is given as follows: $\boldsymbol{\theta}_j = \boldsymbol{\theta}_h$ if $[t_{j-1}, t_j)$ is included in the hot span and $\boldsymbol{\theta}_j = \boldsymbol{\theta}_n$ if $[t_{j-1}, t_j)$ is included in the normal span, i.e., $\tilde{\mathcal{L}}(\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_N; \mathcal{D}(0,T)) = \mathcal{L}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_h; \mathcal{D}(0,T), S^*)$. Thus, we introduce modification vectors, $\boldsymbol{\vartheta}_1, \cdots, \boldsymbol{\vartheta}_N$, defined by $\boldsymbol{\vartheta}_j = \boldsymbol{\theta}_h - \hat{\boldsymbol{\theta}}$ if $[t_{j-1}, t_j)$ is included in the hot span and $\boldsymbol{\vartheta}_j = \boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}$ if $[t_{j-1}, t_j)$ is included in the normal span. Let $\Delta\mathcal{L}$ be $\mathcal{L}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_h; \mathcal{D}(0,T), S^*) - \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}, \cdots, \hat{\boldsymbol{\theta}}; \mathcal{D}(0,T)) = \mathcal{L}(\hat{\boldsymbol{\theta}} + \boldsymbol{\vartheta}_1, \cdots, \hat{\boldsymbol{\theta}} + \boldsymbol{\vartheta}_N; \mathcal{D}(0,T), S^*) - \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}, \cdots, \hat{\boldsymbol{\theta}}; \mathcal{D}(0,T))$. Then, we can obtain the following first-order Taylor expansion:

$$\Delta\mathcal{L} \approx \sum_{j=1}^{N} \frac{\partial \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}, \cdots, \hat{\boldsymbol{\theta}}; \mathcal{D}(0,T))}{\partial \boldsymbol{\theta}_j} \boldsymbol{\vartheta}_j$$

$$= \sum_{j;\ [t_{j-1},t_j) \subset S^*} \frac{\partial \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}, \cdots, \hat{\boldsymbol{\theta}}; \mathcal{D}(0,T))}{\partial \boldsymbol{\theta}_j} (\boldsymbol{\theta}_h - \hat{\boldsymbol{\theta}}) + \sum_{j;\ [t_{j-1},t_j) \not\subset S^*} \frac{\partial \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}, \cdots, \hat{\boldsymbol{\theta}}; \mathcal{D}(0,T))}{\partial \boldsymbol{\theta}_j} (\boldsymbol{\theta}_n - \hat{\boldsymbol{\theta}}).$$

Moreover, by noting Eq. (13), we obtain the following result:

$$\Delta\mathcal{L} \ \approx \sum_{j;\ [t_{j-1},t_j) \subset S^*} \frac{\partial \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}, \cdots, \hat{\boldsymbol{\theta}}; \mathcal{D}(0,T))}{\partial \boldsymbol{\theta}_j} (\boldsymbol{\theta}_h - \boldsymbol{\theta}_n). \qquad (14)$$

Here note that we can naturally assume that each gradient vector with respect to $\boldsymbol{\theta}_j$ is likely to be parallel to $(\boldsymbol{\theta}_h - \boldsymbol{\theta}_n)$, as shown by arrows in Fig. 3. Therefore, from Eq. (14), by considering the following partial sum for a candidate hot span $S = [T_1, T_2) \in \mathcal{H}$:

$$\boldsymbol{g}(S) \ = \sum_{j;\ [t_{j-1},t_j) \subset S} \frac{\partial \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}, \cdots, \hat{\boldsymbol{\theta}}; \mathcal{D}(0,T))}{\partial \boldsymbol{\theta}_j}. \qquad (15)$$

we can expect that $\|\boldsymbol{g}(S)\|$ is maximized when $S \approx S^*$.

Therefore, we propose the method of detecting the hot span by

$$\hat{S} \ = \ \arg\max_{S \in \mathcal{H}} \|\boldsymbol{g}(S)\|. \qquad (16)$$

In case of the AsIC model,

$$\|\boldsymbol{g}(S)\|^2 \ = \left| \sum_{(u,v) \in E;\ u \in \mathcal{D}_h(S)} \frac{\partial \mathcal{L}(\hat{\boldsymbol{p}}, \hat{\boldsymbol{r}}; \mathcal{D}(0,T))}{\partial p_{u,v}} \right|^2$$

(see Eq. (5)), and in case of the VwV model,

$$\boldsymbol{g}(S) = \frac{\partial \mathcal{L}(\hat{\boldsymbol{w}}; \mathcal{D}(T_1, T_2))}{\partial \boldsymbol{w}}$$

(see Eqs. (9)).

Here note that we can incrementally calculate $\boldsymbol{g}(S)$. More specifically, we can obtain the following formula:

$$\boldsymbol{g}([t_i, t_j)) \ = \ \boldsymbol{g}([t_i, t_{j-1})) + \frac{\partial \tilde{\mathcal{L}}(\hat{\boldsymbol{\theta}}, \cdots, \hat{\boldsymbol{\theta}}; \mathcal{D}(0, T))}{\partial \boldsymbol{\theta}_j} \tag{17}$$

for any $t_i, t_{j-1}, t_j \in \mathcal{T}$ with $t_i < t_{j-1} < t_j$. The computational cost of the proposed method for examining each candidate span is much smaller than the naive method described above. When $|\mathcal{T}|$ is very large, we construct a restricted set of candidate spans $\mathcal{H}_J$ as explained above. We summarize our proposed method below.

1. Maximize $\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}(0, T))$ by using the parameter estimation method.
2. Construct the candidate time set $\mathcal{T}$ and the candidate hot span set $\mathcal{H}$.
3. Detect the hot span $\hat{S}$ by Eq. (16) and output $\hat{S}$.
4. Maximize $\mathcal{L}(\boldsymbol{\theta}_n, \boldsymbol{\theta}_h; \mathcal{D}(0, T), \hat{S})$ by using the parameter estimation method, and output $(\hat{\boldsymbol{\theta}_n}, \hat{\boldsymbol{\theta}_h})$.

Here note that the proposed method requires likelihood maximization by using the parameter estimation method only twice.

6. EXPERIMENTAL EVALUATION

We experimentally investigated how accurately the proposed method can estimate both the hot span and the diffusion parameters for the hot and the normal spans, as well as its efficiency, by comparing it with the naive method using four real world networks.

6.1. Datasets

We used four real large networks which are all bidirectionally connected[6]. The first one is a trackback network of Japanese blogs used in [Kimura et al. 2009]. It has $12,047$ nodes and $79,920$ directed links (the blog network). The second one is a coauthorship network used in [Palla et al. 2005], which has $12,357$ nodes and $38,896$ directed links (the Coauthorship network). The third one is a network derived from the Enron Email Dataset [Klimt and Yang 2004] by extracting the senders and the recipients and linking those that had bidirectional communications. It has $4,254$ nodes and $44,314$ directed links (the Enron network). The fourth one is a network of people that was derived from the "list of people" within Japanese Wikipedia, used in [Kimura et al. 2008], and has $9,481$ nodes and $245,044$ directed links (the Wikipedia network).

6.2. Experimental Settings

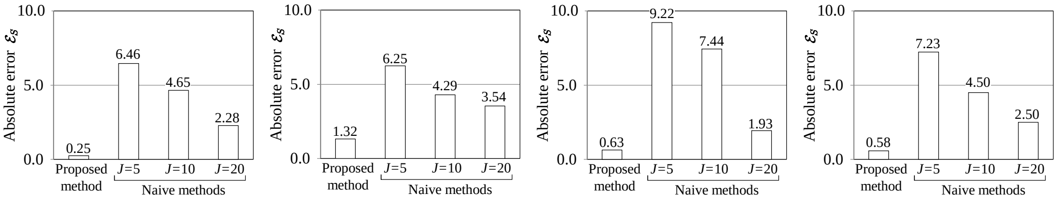
We generated diffusion results using both the AsIC model (for information diffusion evaluation) and the VwV model (for opinion population diffusion evaluation) for each of the above networks under the following settings. As for the AsIC model, we considered $p = 1/\bar{d}$ as the base value of the diffusion probability of each link, where $\bar{d}$ is the mean out-degree of the network. With this base value, for an arbitrary node in the

[6] We wanted to use the real data measured in the real network where there is a known external change, but unfortunately we were not able to find such data. We are still looking for a good dataset that can be used to validate our approach.

network, the expected number of its child nodes that it succeeds to activate becomes approximately equal to one at least at an early phase of the information diffusion. If the diffusion probability is much smaller than the base value, the diffusion process would end up with only a small number of active nodes on the average. On the other hand, if it is much larger, the information rapidly spreads out the entire network and the process finishes at an early phase of the diffusion. Both cases are not appropriate to our aim of investigating the hot span detection, *i.e.*, we need a fair amount of information diffusion taking place around the hot span. Thus, in our experiments, we set the diffusion probability for the normal span, p_n^*, to be a value slightly smaller than the base value, and set the diffusion probability for the hot span, p_h^*, to be three times as large as the p_n^*. As a result, p_n^* and p_h^* are 0.1 and 0.3 for the blog network, 0.2 and 0.6 for the Coauthorship network, 0.05 and 0.15 for the Enron network and 0.02 and 0.06 for the Wikipedia network, respectively. As explained in 3.1, we assumed that the time delay parameter does not change, and fixed its value to be 1 ($r^* = 1$) for all the networks because changing r^* works only for scaling the time axis of the diffusion results. We set the observation period to be $[0, T = 30)$ and the hot span to be $[T_1^* = 10, T_2^* = 20)$ based on the observation on the preliminary experiments. In all we generated 10 information diffusion results using these parameter values, each starting from a randomly selected initial active node for each network.

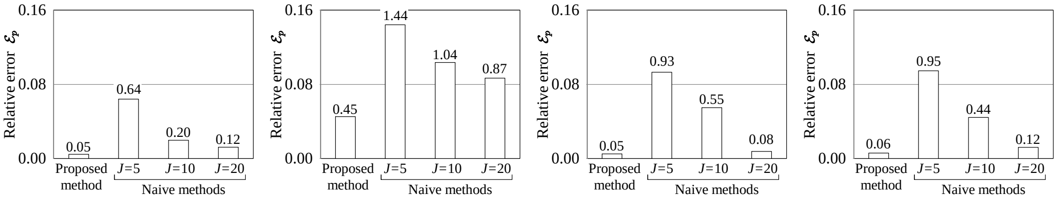
As for the VwV model, for each of the above networks, we generated opinion diffusion results according to the model for three different values of K (the number of opinions), *i.e.*, $K = 2$, 4, and 8, by choosing the top K nodes with respect to node degree ranking as the initial K nodes. We assumed that the value of all the opinions were initially 1.0, *i.e.*, the value-parameters for all the opinions are 1.0 for the normal span, and further assumed that only the value of the first opinion changed to double for the hot span, *i.e.*, the value-parameter of the first opinion is 2.0 and the value-parameters of all the other opinions are 1.0 for the hot span. Again, based on the observation on the preliminary experiments, we set the observation period and the hot span to be $[0, T = 25)$ and $[T_1^* = 10, T_2^* = 15)$, respectively, and generated 10 opinion diffusion results for each network.

We then estimated the hot span $[T_1^*, T_2^*)$ and the diffusion parameters of each model, *i.e.*, the diffusion probabilities p_n^* (for the normal span) and p_h^* (for the hot span) for the AsIC model, and the opinion-value vectors $\boldsymbol{w}_n^*$ (for the normal span) and $\boldsymbol{w}_h^*$ (for the hot span) for the VwV model by the two methods (the proposed and the naive), and compared them in terms of 1) the accuracy of the estimated hot span $\hat{S} = [\hat{T}_1, \hat{T}_2)$, 2) the accuracy of the estimated diffusion parameters, $\hat{p}_n$, $\hat{p}_h$, $\hat{\boldsymbol{w}}_n$, and $\hat{\boldsymbol{w}}_h$, 3) their integrated estimation error, and 4) the computation time. The accuracy of the estimated hot span is measured in the absolute error $\mathcal{E}_S = |\hat{T}_1 - T_1^*| + |\hat{T}_2 - T_2^*|$ for both the AsIC and VwV models. The accuracy of the estimated diffusion parameters is evaluated in the mean relative error, *i.e.*, $\mathcal{E}_p = |\hat{p}_n - p_n|/p_n + |\hat{p}_h - p_h|/p_h$ for the AsIC model, and $\mathcal{E}_{\boldsymbol{w}} = \Sigma_{i=1}^K(|\hat{w}_{n_i} - w_{n_i}^*|/w_{n_i}^* + |\hat{w}_{h_i} - w_{h_i}^*|/w_{h_i}^*)/K$ for the VwV model, where $w_{n_i}^*$ and $w_{h_i}^*$ are values of opinion i for the normal and the hot spans, respectively, and $\hat{w}_{n_i}^*$ and $\hat{w}_{h_i}^*$ are their estimated values. Integrating their estimation errors by $\mathcal{E}_{\boldsymbol{\theta}(t)} = \int_0^T ||\hat{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}^*(t)||_{L1} dt$ allows us to evaluate the estimation ability of each method in a comprehensive manner, where $\boldsymbol{\theta}^*(t)$ and $\hat{\boldsymbol{\theta}}(t)$ is the diffusion parameter vector to be assumed true and its estimation at time t for the corresponding model, respectively. For the proposed method, we adopted $1,000$ as the value of J (the number of candidate time points) for the VwV model, while we used all the possible time points, *i.e.*, $J = N$ for the AsIC model because the number of time points for opinion changes in the VwV model is observed to be much larger than the number of node activation for the AsIC



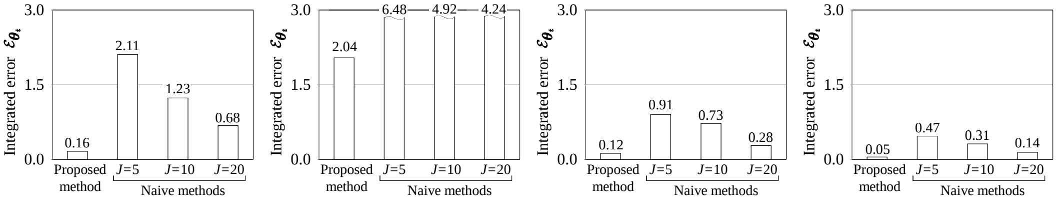
(a) Blog (b) Coauthorship (c) Enron (d) Wikipedia

Fig. 4: Comparison of the accuracy in the estimated hot span for the AsIC model.



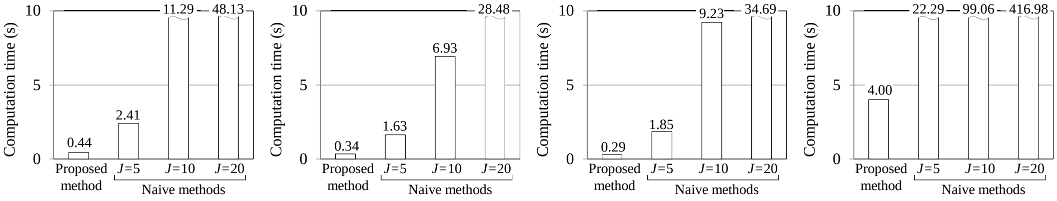
(a) Blog (b) Coauthorship (c) Enron (d) Wikipedia

Fig. 5: Comparison of the accuracy in the estimated diffusion probability for the AsIC model.



(a) Blog (b) Coauthorship (c) Enron (d) Wikipedia

Fig. 6: Comparison of the integrated estimation error for the AsIC model.



(a) Blog (b) Coauthorship (c) Enron (d) Wikipedia

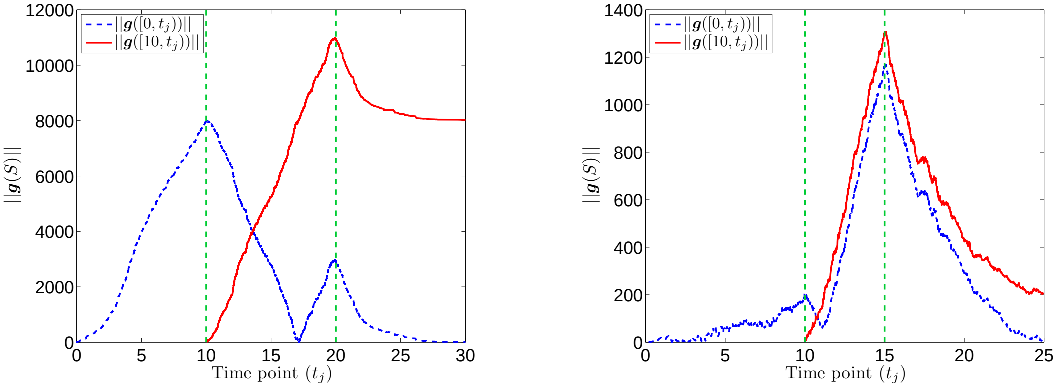
Fig. 7: Comparison of the computation time for the AsIC model.

model for the same period of time. For the naive method, we tested three cases of $J = 5$, 10, and 20 for both the models. Both the proposed and the naive methods were tested on each diffusion result for each model mentioned above on a PC with Intel Core 2 Duo 3GHz, and the results were averaged over the ten independent trials for each network.

6.3. Results for AsIC Model

Figures 4 to 7 summarize the results for the AsIC model. Figure 4 shows the accuracy of $\hat{S}$ in the absolute error $\mathcal{E}_S$ defined above. We see that the proposed method achieves a good accuracy, much better than the naive method for every network. As expected, $\mathcal{E}_S$ for the naive method decreases as J becomes larger. But, even in the best case ($J = 20$), its average error is about 3 to 10 times larger than that of the proposed method. Figure 5 shows the accuracy of $\hat{p}_n$ and $\hat{p}_h$ in the relative error $\mathcal{E}_p$. Here again, the average relative error for the naive method decreases as J becomes larger. However, even in the best case ($J = 20$), it is about 2 to 3 times larger than that of the proposed method. We note that the average errors for the Coauthorship network are relatively large. This is because the number of active nodes within the normal span was relatively small for this network. Figure 6 shows the integrated estimation error given by $\mathcal{E}_{\boldsymbol{\theta}(t)}$, which supplements our insights derived from the above results. For example, although the relative error of the estimated diffusion probabilities of the naive method ($J = 20$) is less than twice as big as the proposed method for the Enron network, its value of $\mathcal{E}_{\boldsymbol{\theta}(t)}$ becomes more than twice of the proposed method by considering the estimation error of the hot span. Overall, even in the best case of the naive method ($J = 20$), its integrated estimation error is about 2 to 4 times larger than that of the proposed method. Figure 7 shows the computation time. It is clear that the proposed method is much faster than the naive method. The significant difference is attributed to the difference in the number of runs of the EM-like algorithm. The proposed method executes the EM-like algorithm only twice: steps 1 and 4 in the algorithm (see Section 5.2). On the other hand, the naive method has to execute the EM-like algorithm once for every single candidate hot span $S \in \mathcal{H}_J$ which is $|\mathcal{H}_J| = J(J-1)/2$ times (see Section 5.1). Indeed, the computation time of the naive method for $J = 5$ is about 5 times larger for every network, which is consistent with the fact that $|\mathcal{H}_5| = 10$. This relation roughly holds also for the other two cases ($J = 10$ and $J = 20$). This means that even if the naive method could achieve a good accuracy by setting J to a sufficiently large value, it would require unacceptable computation time for such a large J. Overall, the proposed algorithm is about 3 times more accurate in the fastest case for the naive method (in the case of the Coauthorship network under $J = 5$) and about 100 times faster in its most accurate case (in the case of the Wikipedia network under $J = 20$). Finally, we illustrate the actual behavior of $\|\boldsymbol{g}(S)\|$ derived from an information diffusion result for the blog network under the AsIC model in Fig. 8a, where $\|\boldsymbol{g}(S)\|$ is depicted as a function of the ending point t_j of S when its starting point is fixed to a certain value. We can see the blue broken curve showing $\|\boldsymbol{g}([0, t_j))\|$ has two peaks at around $t_j = 10$ and $t_j = 20$, which are the starting and ending points of the true hot span, respectively. This means that the sign of $\partial\mathcal{L}(\hat{\boldsymbol{p}}, \hat{\boldsymbol{r}}; \mathcal{D}(0, T))/\partial p_{u,v}$ reversed at these time points as explained in Section 5.2 [7]. Thus, the red solid curve showing $\|\boldsymbol{g}([10, t_j))\|$ has only one peak at around $t_j = 20$, which is the global maximum among all the possible $\|\boldsymbol{g}(S)\|$. Thanks to Eq.(17), the proposed method can efficiently calculate the behavior of $\|\boldsymbol{g}(S)\|$, and thus can find out the hot span more accurately and more efficiently than the naive method does.

In summary, we can say that the proposed method can detect and estimate the hot span and diffusion probabilities for the AsIC model much more accurately and efficiently compared with the naive method.



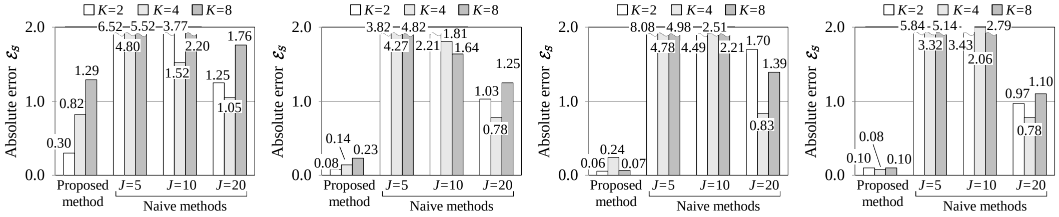
(a) AsIC Model (b) VwV Model

Fig. 8: Change of $||\boldsymbol{g}(S)||$ when given a fixed starting point of a time span S for a diffusion result retrieved from the blog network under the respective experimental setting for each information diffusion model.

6.4. Results for Voter Model

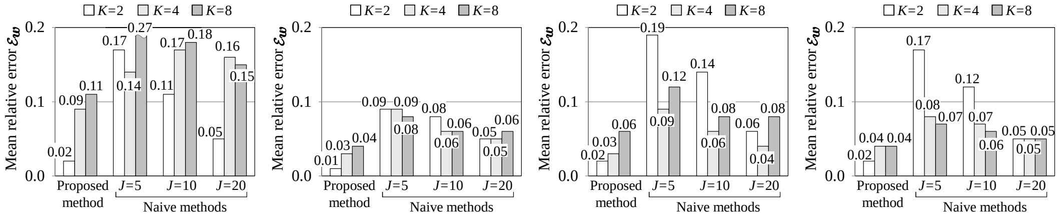
Figures 9 to 12 show the experimental results for the VwV model. Similarly to the results for the AsIC model, from these results, we can find that the proposed method is much more accurate than the naive method for every network. Again, the average error for the naive method decreases as J becomes larger. But, even in the best case for the naive method ($J = 20$), its average error in the estimation of the hot span is maximum about 30 times larger than that of the proposed method (in the case of the Enron network under $K = 2$) as shown in Fig. 9, and it is maximum about 6 times larger in the estimation of opinion-values (in the case of the Coauthorship network under $K = 2$) as shown in Fig. 10. Figure 11 shows that the proposed method is better than the naive method in the integrated estimation accuracy for every case. It is noted that the naive method needs much longer computation time to achieve these best accuracies than the proposed method as shown in Fig. 12 despite that the number of candidate time points for the naive method is 50 times smaller. Indeed, it is about 20 times longer in the case of the Enron network under $K = 2$, about 13 times longer in the case of the Coauthorship network under $K = 2$, and maximum about 95 times longer for the whole results (in the case of the Enron network under $K = 8$). Overall, the proposed method is about 7 times more accurate in the fastest case for the naive method (in the case of the blog network under $K = 2$ and $J = 5$) and about 13 times faster in its most accurate case (in the case of the Coauthorship network under $K = 2$ and $J = 20$). Figure 8b shows the behavior of $||\boldsymbol{g}(S)||$ derived from an opinion diffusion result for the blog network under the VwV model. Similarly to the case of the AsIC model, it is found that the blue broken curve showing $||\boldsymbol{g}([0,t_j))||$ has two peaks at around $t_j = 10$ and $t_j = 15$, which are the starting and ending points of the true hot span, respectively. In this case, the red solid curve starting from $t_j = 10$ has only one peak at around $t_j = 15$, which becomes the global maximum among all the possible $||\boldsymbol{g}(S)||$. The proposed method can find out the time span that results in the global maximum from a set of the candidate time points efficiently for the VwV model, too.

[7] Since in this case the partial derivative is a scalar, it suffices to say its sign.



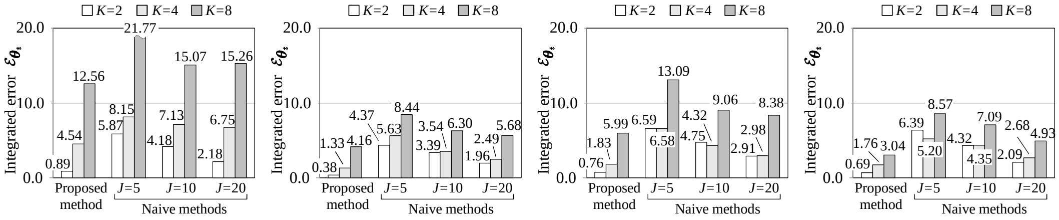
(a) Blog (b) Coauthorship (c) Enron (d) Wikipedia

Fig. 9: Comparison of the accuracy in the estimated hot span for the VwV model.



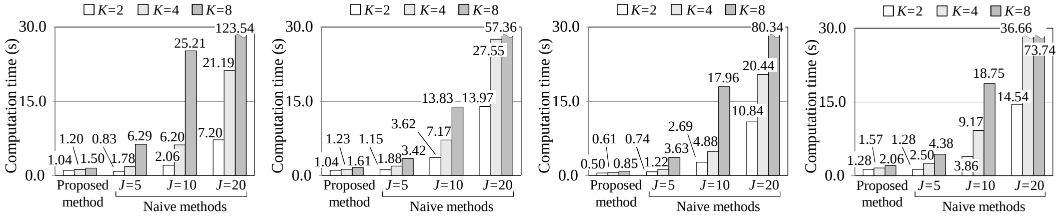
(a) Blog (b) Coauthorship (c) Enron (d) Wikipedia

Fig. 10: Comparison of the accuracy in the estimated opinion-value vector for the VwV model.



(a) Blog (b) Coauthorship (c) Enron (d) Wikipedia

Fig. 11: Comparison of the integrated estimation error for the VwV model.



(a) Blog (b) Coauthorship (c) Enron (d) Wikipedia

Fig. 12: Comparison of the computation time for the VwV model.

From these results, it can be concluded that the proposed method is able to detect and estimate the hot span and opinion-values for the VwV model much more accurately and efficiently compared with the naive method.

7. DISCUSSION

The results in the previous section indicates that the proposed approach works as intended to both AsIC and VwV diffusion models. Although we believe that the approach is generic, it has yet to be verified whether the approach is applicable to any other model in so far as it is formulated as a probabilistic diffusion model.

We placed a simplifying constraint that the parameters $p_{u,v}$ and $r_{u,v}$ of the AsIC model do not depend on link (u, v), *i.e.*, $p_{u,v} = p$, $r_{u,v} = r$ $(\forall(u, v) \in E)$, by focusing on single topic diffusion sequences. Our previous experiments [Saito et al. 2009b; 2010a; 2010b] give some evidences which support the validity of this constraint. We examined $7,356$ diffusion sequences for a real blogroll network containing $52,525$ bloggers and $115,552$ blogroll links, and have experimentally confirmed that the diffusion and time-delay parameters that were learned from different diffusion sequences belonging to the same topic were quite similar for most of the topics. This observation naturally suggests that people behave quite similarly for the same topic. On the other hand, our recent study indicates that these parameters can be learned by assuming their functional dependency on the neighboring node attributes [Saito et al. 2011b]. We can extend this approach to augment the attributes to include the node independent external factor. This way the uniformity assumption can be removed. We have considered only the AsIC as a model of general information diffusion, but it is straightforward to apply the same technique to the AsLT model [Saito et al. 2010b] and the SIS (susceptible/infectious/susceptible) versions of both the models in which each node is allowed to be activated multiple times.

The change pattern we used is also very simple. We assumed that the parameters of all nodes and links change instantaneously and simultaneously in the same degree and stay the same during a given hot span. We can assume a more intricate problem setting such that $p_{u,v}$ (for AsIC), w_u (for VwV) and $r_{u,v}$ (for both) change for multiple distinct hot spans and the shape of change pattern $p_{u,v}$ and w_u are not necessarily rect-linear. One possible extension is to approximate the pattern of any shape by J pairs of time interval each with its corresponding $p_{u,v,j}$ and $w_{u,j}$, *i.e.*, $Z_J = \{(p_j, [t_{j-1}, t_j));\ j = 1, \cdots J\}$ $(t_0 = 0, t_J = \infty)$ and use a divide-and-conquer type greedy recursive partitioning, still employing the derivative of the likelihood function as the main measure for search. For brevity we drop the u, v dependency and consider only the AsIC model. More specifically, we first initialize $Z_1 = \{(\hat{p}_1, ([0, \infty))\}$ where $\hat{p}_1$ is the maximum likelihood estimator, and search for the first change time point t_1, which we expect to be the most distinguished one, by maximizing $\|\boldsymbol{g}(S)\|$ that uses $\hat{p}_1$ as $\hat{\theta}$ for the whole span $[0, \infty)$.[8] We recursively perform this operation J times by fixing the previously determined change points. When to stop can be determined by a statistical criterion such as AIC or MDL. This algorithm requires parameter optimization J times. Figure 13 is one of the preliminary results obtained for two distinct rect-linear patterns using five sequences in case of the blog network. MDL is used as the stopping criterion. The change pattern of p is almost perfectly detected with respect to both p_j and t_j ($J = 5$). We might further want to introduce some stochastic natures into the model for some external factors that affect parameter changes reflecting the fact that each individual's response to the external factors is different, *i.e.*, some people respond quickly and others slowly.

The change we considered is only in the time domain and we assumed that there is no spatially local change. We can consider a more general setting, *i.e.*, spatio-temporal change in parameter values. We need a more elaborate algorithm to cope with this extension but the basic approach of using the first derivative of the likelihood function

[8] Note that the total sum of $\boldsymbol{g} = 0$.

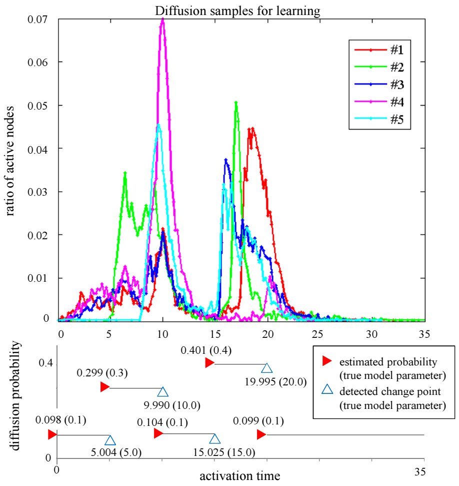


Fig. 13: Information diffusion in the blog network with two hot spans for the AsIC model.

remains valid. We assumed that the network structure is stationary although we introduced the change in the parameter value. The model we used does not account for the structure change by itself. However, once the structure change is known, *i.e.*, addition/deletion of nodes and links at each time instance, it is straightforward to apply the proposed algorithm to these changes because the dynamics of a node is determined by the interaction with its neighbors, *i.e.*, local structure of the network.

8. CONCLUSION

In this paper, we addressed the problem of detecting changes in behavior of information diffusion over a social network which is caused by changes in unknown external factors from a limited amount of observed diffusion sequences in a retrospective setting. The information diffusion process is described by a probabilistic model with some parameters that characterize the behavior, and the change in unknown external factors is assumed to be effectively reflected in changes in the parameter values in the model. We called the period where the parameter takes anomalous values as “hot span” and the rest as “normal span”, and the problem is reduced to detecting the hot span, *i.e.*, identifying the time window where the parameter value is anomalous and estimating the parameter values both in the hot and normal spans. We solved this problem by searching the time window that maximizes the likelihood of generating the observed information diffusion sequences. Our main contribution is that we devised a very efficient general iterative search algorithm which is robust and applicable to a wide class of probabilistic information diffusion models. The algorithm uses the first derivative of the likelihood with respect to the parameters, uses it in the window search (outer loop) and avoids parameter value optimization during the search (inner loop). It only needs to estimate the parameter value twice (at the first and the final steps of the search). This is in contrast to the naive learning algorithm which has to iteratively update the pattern boundaries (outer loop), each requiring the parameter value optimization to maximize the likelihood for the candidate window (inner loop), which is very inefficient and totally unacceptable. We showed that the algorithm works satisfactorily well for two instances of the probabilistic information diffusion model which has different characteristics: asynchronous independent cascade (AsIC) model as a model of information push style and value-weighted voter (VwV) model as a model of information pull style. The AsIC is the model for general information diffusion with binary states and the parameter to detect its change is diffusion probability and the VwV is the model for opinion formation with multiple states and the parameter to detect its change is opinion value. The results tested on these two models using four real world network structures and a single rect-linear change confirmed that the algorithm is robust enough and can efficiently identify the correct change pattern of the parameter values. Comparison with the naive method that finds the best combination of change boundaries by an exhaustive search through a set of randomly selected boundary candidates showed that the proposed algorithm far outperforms the native method both in terms of accuracy (about 3 times more accurate for the AsIC model and about 7 times accurate for the VwV model in the fastest case for the naive method) and computation time (about 100 times faster for the AsIC model and about 13 times faster for the VwV model in the most accurate case for the naive method). The problem setting we assumed in this paper is very simple, but we expect that the proposed method can be easily extended to solve more intricate problems. We showed one possible direction and the preliminary result obtained for two rect-linear shape hot spans was very promising. Our immediate future work is to evaluate our method using real world information diffusion samples with hot spans, as well as to deal with spatio-temporal hot span detection problems using more appropriate stochastic models under a similar problem solving framework.

APPENDIX

A. ESTIMATION ALGORITHM FOR ASIC MODEL

We briefly describe the estimation algorithm of parameters p and r for the AsIC model from a sequence of observed data $\mathcal{D}(0,T)$ (see [Saito et al. 2009b; 2010a] for more details).

We employ an EM-like algorithm. Let $\bar{p}$ and $\bar{r}$ be the current estimates of p and r. Using Eqs. (1) and (2), we define $\bar{\alpha}_{u,v}$ and $\bar{\beta}_{u,v}$ as follows:

$$\bar{\alpha}_{u,v} = \frac{\mathcal{X}_{u,v}(\bar{p},\bar{r})/\mathcal{Y}_{u,v}(\bar{p},\bar{r})}{\sum_{z\in\mathcal{A}_v}\mathcal{X}_{z,v}(\bar{p},\bar{r})/\mathcal{Y}_{z,v}(\bar{p},\bar{r})}$$

$$\bar{\beta}_{u,v} = \frac{\bar{p}\exp(-\bar{r}(t_v - t_u))}{\mathcal{Y}_{u,v}(\bar{p},\bar{r})}$$

The update formulas of $\boldsymbol{p}$ and $\boldsymbol{r}$ are as follows:

$$p = \frac{\sum_{v\in\mathcal{D}}\sum_{u\in\mathcal{A}_v}\left(\bar{\alpha}_{u,v} + (1-\bar{\alpha}_{u,v})\bar{\beta}_{u,v}\right)}{|\{(u,v)\in E;\ u\in\mathcal{D}\}|}$$

$$r = \frac{\sum_{v\in\mathcal{D}}\sum_{u\in\mathcal{A}_v}\bar{\alpha}_{u,v}}{\sum_{v\in\mathcal{D}}\sum_{u\in\mathcal{A}_v}\left(\bar{\alpha}_{u,v} + (1-\bar{\alpha}_{u,v})\bar{\beta}_{u,v}\right)(t_v - t_u)}.$$

B. ESTIMATION ALGORITHM FOR VWV MODEL

We briefly describe the estimation algorithm of parameter vector $\boldsymbol{w}$ for the VwV model from an observed data $\mathcal{D}(0,T)$ (see [Kimura et al. 2010b] for more details). As mentioned in Subsection 3.2, the opinion dynamics for the VwV model is invariant to positive scaling of $\boldsymbol{w}$. Thus, we transform the parameter vector $\boldsymbol{w}$ by $\boldsymbol{w} = \boldsymbol{w}(\boldsymbol{z})$, where

$$\boldsymbol{w}(\boldsymbol{z}) = (\exp(z_1),\cdots,\exp(z_{K-1}),1), \quad \left(\boldsymbol{z} = (z_1,\cdots,z_{K-1})\in\mathbf{R}^{K-1}\right).$$

Namely, our problem is to estimate the value of $\boldsymbol{z}$ that maximize $\mathcal{L}(\boldsymbol{w}(\boldsymbol{z});\mathcal{D}(0,T))$.

Then, for any $i, j \in \{1, \cdots, K-1\}$, we obtain

$$\frac{\partial \mathcal{L}(\boldsymbol{w}(\boldsymbol{z}); \mathcal{D}(0,T))}{\partial z_i} = \sum_{(v,t,k) \in \mathcal{C}(0,T)} (\delta_{k,i} - q_i(t,v)),$$

$$\frac{\partial^2 \mathcal{L}(\boldsymbol{w}(\boldsymbol{z}); \mathcal{D}(0,T))}{\partial z_i \, \partial z_j} = \sum_{(v,t,k) \in \mathcal{C}(0,T)} (q_i(t,v)\, q_j(t,v) - \delta_{i,j}\, q_i(t,v)),$$

where $\delta_{i,j}$ is the Kronecker's delta, and

$$q_i(t,v) = \frac{n_i(t,v) \exp(z_i)}{n_K(t,v) + \sum_{\ell=1}^{K-1} n_\ell(t,v) \exp(z_\ell)}.$$

We can show that the Hessian matrix $\left(\partial^2 \mathcal{L}(\boldsymbol{w}(\boldsymbol{z}); \mathcal{D}(0,T)) / \partial z_i \partial z_j\right)$ is negative semi-definite. Hence, by solving the equations $\partial \mathcal{L}(\boldsymbol{w}(\boldsymbol{z}); \mathcal{D}(0,T)) / \partial z_i = 0$, $(i = 1, \cdots, K-1)$, we can find the value of $\boldsymbol{z}$ that maximizes $\mathcal{L}(\boldsymbol{w}(\boldsymbol{z}); \mathcal{D}(0,T))$. We employed a standard Newton Method in our experiments.

REFERENCES

CASTELLANO, C., MUNOZ, M. A., AND PASTOR-SATORRAS, R. 2009. Nonlinear q-voter model. *Physical Review E 80*, 041129.

CHEN, W., WANG, C., AND WANG, Y. 2010a. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010)*. 1029–1038.

CHEN, W., YUAN, Y., AND ZHANG, L. 2010b. Scalable influence maximization in social networks under the linear threshold model. In *Proceedings of the 10th IEEE International Conference on Data Mining (ICDM 2010)*. 88–97.

CRANDALL, D., COSLEY, D., HUTTENLOCNER, D., KLEINBERG, J., AND SURI, S. 2008. Feedback effects between similarity and sociai infiuence in online communities. In *Proceedings of KDD 2008*. 160–168.

DOMINGOS, P. 2005. Mining social networks for viral marketing. *IEEE Intelligent Systems 20*, 80–82.

EVEN-DAR, E. AND SHAPRIA, A. 2007. A note on maximizing the spread of influence in social networks. In *Proceedings of WINE 2007*. 281–286.

GOLDENBERG, J., LIBAI, B., AND MULLER, E. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters 12*, 211–223.

GOMEZ-RODRIGUEZ, M., LESKOVEC, J., AND KRAUSE, A. 2010. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010)*. 1019–1028.

GRUHL, D., GUHA, R., LIBEN-NOWELL, D., AND TOMKINS, A. 2004. Information diffusion through blogspace. *SIGKDD Explorations 6*, 43–52.

HOLME, P. AND NEWMAN, M. E. J. 2006. Nonequilibrium phase transition in the coevolution of networks and opinions. *Physical Review E 74*, 056108.

KEMPE, D., KLEINBERG, J., AND TARDOS, E. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*. 137–146.

KIMURA, M., SAITO, K., AND MOTODA, H. 2008. Minimizing the spread of contamination by blocking links in a network. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*. 1175–1180.

KIMURA, M., SAITO, K., AND MOTODA, H. 2009. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data 3*, 9:1–9:23.

KIMURA, M., SAITO, K., NAKANO, R., AND MOTODA, H. 2010a. Extracting influential nodes on a social network for information diffusion. *Data Min. Knowl. Disc. 20*, 70–97.

KIMURA, M., SAITO, K., OHARA, K., AND MOTODA, H. 2010b. Learning to predict opinion share in social networks. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10)*. 1364–1370.

KIMURA, M., SAITO, K., OHARA, K., AND MOTODA, H. 2011. Detecting anti-majority opinionists using value-weighted mixture voter model. In *Proceedings of the 14th International Conference on Discovery Science (DS 2011)*. LNAI 6926, 150–164.

KLEINBERG, J. 2002. Bursty and hierarchical structure in streams. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*. 91–101.

KLIMT, B. AND YANG, Y. 2004. The enron corpus: A new dataset for email classification research. In *Proceedings of the 2004 European Conference on Machine Learning (ECML'04)*. 217–226.

LESKOVEC, J., ADAMIC, L. A., AND HUBERMAN, B. A. 2006. The dynamics of viral marketing. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06)*. 228–237.

LIGGETT, T. M. 1999. *Stochastic interacting systems: contact, voter, and exclusion processes*. Spriger, New York.

MYERS, S. A. AND LESKOVEC, J. 2010. On the convexity of latent social network inference. In *Proceedings of Neural Information Processing Systems (NIPS)*.

NEWMAN, M. E. J. 2003. The structure and function of complex networks. *SIAM Review 45*, 167–256.

NEWMAN, M. E. J., FORREST, S., AND BALTHROP, J. 2002. Email networks and the spread of computer viruses. *Physical Review E 66*, 035101.

OHARA, K., SAITO, K., KIMURA, M., AND MOTODA, H. 2011. Efficient detection of hot span in information diffusion from observation. In *Proceedings of the IJCAI Workshop on Link Analysis in Heterogeneous Information Networks (HINA 2011)*. arXiv: 1110.2659.

PALLA, G., DERÉNYI, I., FARKAS, I., AND VICSEK, T. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature 435*, 814–818.

SAITO, K., KIMURA, M., NAKANO, R., AND MOTODA, H. 2009a. Finding influential nodes in a social network from information diffusion data. In *Proceedings of the International Workshop on Social Computing and Behavioral Modeling (SBP09)*. 138–145.

SAITO, K., KIMURA, M., OHARA, K., AND MOTODA, H. 2009b. Learning continuous-time information diffusion model for social behavioral data analysis. In *Proceedings of the 1st Asian Conference on Machine Learning (ACML2009)*. LNAI 5828, 322–337.

SAITO, K., KIMURA, M., OHARA, K., AND MOTODA, H. 2010a. Behavioral analyses of information diffusion models by observed data of social network. In *Proceedings of the 2010 International Conference on Social Computing and Behavioral Modeling (SBP10)*. 149–158.

SAITO, K., KIMURA, M., OHARA, K., AND MOTODA, H. 2010b. Selecting information diffusion models over social networks for behavioral analysis. In *Proceedings of the 2010 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2010)*. LNAI 6323, 180–195.

SAITO, K., KIMURA, M., OHARA, K., AND MOTODA, H. 2011a. Detecting changes in opinion value distribution for voter model. In *Proceedings of the 4th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction (SBP2011)*. LNAI 6389, 89–96.

SAITO, K., OHARA, K., YAMAGISHI, Y., KIMURA, M., AND MOTODA, H. 2011b. Learning diffusion probability based on node attributes in social networks. In *Proceedings of the 19th International Symposium on Methodologies for Intelligent Systems (ISMI S2011)*. LNAI 6804, 153–162.

SOOD, V. AND REDNER, S. 2005. Voter model on heterogeneous graphs. *Physical Review Letters 94*, 178701.

SWAN, R. AND ALLAN, J. 2000. Automatic generation of overview timelines. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*. 49–56.

WATTS, D. J. 2002. A simple model of global cascades on random networks. *Proceedings of National Academy of Science, USA 99*, 5766–5771.

WATTS, D. J. AND DODDS, P. S. 2007. Influence, networks, and public opinion formation. *Journal of Consumer Research 34*, 441–458.

WU, F. AND HUBERMAN, B. A. 2008. How public opinion forms. In *Proceedings of WINE 2008*. 334–341.

YAMAGISHI, Y., SAITO, K., OHARA, K., KIMURA, M., AND MOTODA, H. 2011. Learning attribute-weighted voter model over social networks. In *Proceedings of the 3rd Asian Conference on Machine Learning (ACML 2011), to appear*. JMLR Workshop and Conference Proceedings.

YANG, H., WU, Z., ZHOU, C., ZHOU, T., AND WANG, B. 2009. Effects of social diversity on the emergence of global consensus in opinion dynamics. *Physical Review E 80*, 046108.

Received June 2011; revised November 2011; accepted December 2011

Network Analysis of Three Twitter Functions: Favorite, Follow and Mention

Shoko Kato[1], Akihiro Koide[1], Takayasu Fushimi[1],
Kazumi Saito[1], and Hiroshi Motoda[2]

[1] School of Management and Information, University of Shizuoka,
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan,
{b09032,j11103,j11507,k-saito}@u-shizuoka-ken.ac.jp
[2] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We analyzed three functions of Twitter (Favorite, Follow and Mention) from network structural point of view. These three functions are characterized by difference and similarity in various measures defined in directed graphs. Favorite function can be viewed by three different graph representations: a simple graph, a multigraph and a bipartite graph, Follow function by one graph representation: a simple graph, and Mention function by two graph representations: a simple graph and a multigraph. We created these graphs from three real world twitter data and found salient features characterizing these functions. Major findings are a very large connected component for Favorite and Follow functions, scale-free property in degree distribution and predominant mutual links in certain network motifs for all three functions, freaks in Gini coefficient and two clusters of popular users for Favorites function, and a structure difference in high degree nodes between Favorite and Mention functions characterizing that Favorite operation is much easier than Mention operation. These finding will be useful in building a preference model of Twitter users.

1 Introduction

Grasping and controlling preference, tendency, or trend of the consuming public is one of the important factors to achieve economical success. Accordingly, it is vital to collect relevant data, analyze them and model user preference. However, quantifying preference is very difficult to achieve and finding useful measures from the network structure is crucial. The final goal of this work is to find such measures, characterize their relations and build a reliable user preference model based on these measures from the available data. As the very first step, we focus on Twitter data and analyzes the user behavior of three functions (Favorite, Follow and Mention) of Twitter [1] from the network structural point of view, i.e., by using various measures that have been known useful in the graph theory and

[1] http://twitter.com/

identifying characteristic features (difference and similarity) of these measures for these functions.

User behavior of these three functions are represented by different directed graphs. Favorite function can be viewed by three different graph representations: a simple graph, i.e., single edge from a Favorer to a Favoree, a multigraph, i.e., multiple edges from a Favorer to a Favoree, and a bipartite graph, i.e., single edge from a Favorer to a Favoree treating a user with both a Favorer and a Favoree as two separate nodes. Likewise, Follow function can be viewed by one graph representation: a simple graph, i.e., single edge from a Follower and a Followee, and Mention function can be viewed by two different graphs: a simple graph, i.e. single edge from a Mentioner (sender) to a Mentionee (receiver) and a multigraph, i.e. multiple edges from a Mentioner to a Mentionee. We have created these networks from three different Twitter logs (called ”Favorites network”, ”Followers network”, and ”Mentions network”) and used several different measures, e.g. in-degree, out-degree, multiplicity, Gini coefficient, etc. Extensive experiments were performed and several salient features were found. Major findings are that 1) Favorites and Followers networks have a very large connected component but Mentions network is not, 2) all the three networks (both simple and multiple) have the scale-free property in degree distribution, 3) all three networks (simple) have predominant three-node motifs having mutual links, 4) Favorites network have freaks in Gini coefficient (one of the measures), 5) Favorites network have two clusters of popular users, and 6) Favorites and Mentions networks differ in structure for high degree nodes reflecting that Favorite operation is much easier than Mentions operation. We analyse simple graph and bipartite graph with conventional methods, and multigraph with new proposal method using Gini coefficient, the index which measures the inequality among values of a frequency distribution. Twitter, a microblogging service, has attracted a great deal of attention and various properties have already been obtained [3] [4], but to our knowledge, there have been no work to analyze the user behavior from network structural point of view. We believe that the work along this line will be useful in understanding the user behavior and helps building a preference model of Twitter users.

The paper is organized as follows. We briefly explain the various measures we adopted in our analysis in 2, three networks (Favorite, Follow, and Mention) in 3. Then we report the experimental results in 4 and provide some discussions regarding our observations in 5. We end this paper by summarizing the major finding and mentioning the future work in 6.

2 Analysis Methods

According to [1], we define the structure of a network as a graph. A graph $G = (V, E)$ consists of a set V of nodes (vertices) and a set E of links (edges) that connect pairs of nodes. Note that in our Favorites, Followers or Mentions network, a node corresponds to a Twitter user, and a link corresponds to favoring, following, or mentioning between a pair of users. If two nodes are connected by a link, they are adjacent and we call them neighbors. In directed graphs, each directed link has an origin (source) and a destination (target). A link with origin $u \in V$ and destination $v \in V$ is represented by an ordered pair (u, v). A directed graph $G = (V, E)$ is called a bipartite graph, if V is divided into to two parts, V_x and V_y, where $V = V_x \cup V_y$, $V_x \cap V_y = \emptyset$, and $E \subset \{(u, v); u \in V_x, v \in V_y$. In directed graphs, we may allow the link set E to contain the same link several times, i.e., E can be a multiset. If a link occurs several times in E, the copies of that link are called parallel links. Graphs with parallel links are also called multigraphs. A graph is called simple, if each of its links is contained in E only once, i.e., if the graph does not have parallel links. In what follows, we describe our analysis methods for each type of graphs.

2.1 Methods for Simple Graph

A graph $G' = (V', E')$ is a subgraph of the graph $G = (V, E)$ if $V' \in V$ and $E' \in E$. It is an induced subgraph if E' contains all links $e \in E$ that connect nodes in V'. A directed graph $G = (V, E)$ is strongly connected if there is a directed path from every node to every other node. A strongly connected component of a directed graph G is an induced subgraph that is strongly connected and maximal. A bidirected graph $\tilde{G} = (V, \tilde{E})$ is constructed from a directed graph $G = (V, E)$ by adding counterparts of the unidirected links, i.e., $\tilde{E} = E \cup \{(v, u); (u, v) \in E\}$. A weakly connected component of a directed graph G is an induced subgraph from V' obtained as a strongly connected component of the bidirected graph $\tilde{G}$. We analyze the structure of our networks in terms of the connectivity using these notions.

In a directed graph $G = (V, E)$, the out-degree of $v \in V$, denoted by $d^+(v)$, is the number of links in E that have origin v. The in-degree of $v \in V$, denoted by $d^-(v)$, is the number of links with destination v. The average degree d is calculated by

$$d = \frac{1}{|V|} \sum_{v \in V} d^-(v) = \frac{1}{|V|} \sum_{v \in V} d^+(v) = \frac{|E|}{|V|}. \tag{1}$$

Here $| \cdot |$ stands for the number of elements for a given set. The correlation between in- and out-degree, denoted by c, is calculated by

$$c = \frac{\sum_{v \in V} (d^-(v) - d)(d^+(v) - d)}{\sqrt{\sum_{v \in V} (d^-(v) - d)^2} \sqrt{\sum_{v \in V} (d^+(v) - d)^2}}. \tag{2}$$

On the other hand, the in-degree distribution $id(k)$ and the out-degree distribution $od(k)$ with respect to degree k are respectively defined by

$$id(k) = |\{v \in V; d^-(v) = k\}|, \quad od(k) = |\{v \in V; d^+(v) = k\}|. \tag{3}$$

We analyze the statistical properties of these degree distributions.

Network motifs are defined as patterns of interconnections occurring in graphs at numbers that are significantly higher than those in randomized graphs. In our analysis, we focus on three-node motifs patterns and Figure 1 shows all thirteen

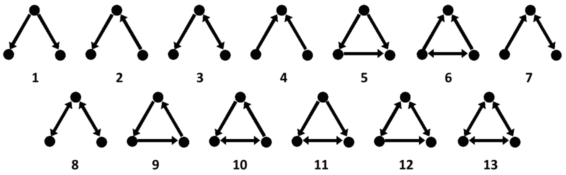


Fig. 1: Network motifs patterns

types of three-node connected subgraphs (motifs patterns). According to [5], we also use randomized graphs, each node of which has the same in-degree and out-degree as the corresponding node has in the real network [6]. A significance level of each motifs pattern i is evaluated by its z-score z_i, i.e.,

$$z_i = \frac{f_i - J^{-1} \sum_{j=1}^{J} g_{j,i}}{\sqrt{J^{-1} \sum_{j=1}^{J} (f_i - J^{-1} \sum_{j=1}^{J} g_{j,i})^2}}, \qquad (4)$$

where J is the number of randomized graphs used for evaluation, and f_i and $g_{j,i}$ denote the numbers of occurrences of motifs pattern i in the real graph and the j-th randomized graph, respectively. By this motifs analysis, we attempt to uncover the basic building blocks of our networks.

2.2 Visualization of Bipartite Graph

We can construct a bipartite graph from a directed graph by setting $V_x = \{u; (u,v) \in E\}$ and $V_y = \{v; (u,v) \in E\}$, and regarding that any element in V_x is different from any element in V_y. Further, according to [2], we describe a bipartite graph visualization method for our analysis. For the sake of technical convenience, each set of the nodes, V_x and V_y, is identified by two different series of positive integers, i.e., $V_x = \{1, \cdots, m, \cdots, M\}$ and $V_y = \{1, \cdots, n, \cdots, N\}$. Here M and N are the numbers of the nodes in V_x and V_y , i.e., $|V_x| = M$ and $|V_y| = N$, respectively. Then, the $M \times N$ adjacency matrix $\mathbf{A} = \{a_{m,n}\}$ is defined by setting $a_{m,n} = 1$ if $(m,n) \in E$; $a_{m,n} = 0$ otherwise. The L-dimensional embedding position vectors are denoted by $\mathbf{x}_m$ for the node $m \in V_x$ and $\mathbf{y}_n$ for the node $n \in V_y$. Then we can construct $M \times L$ and $N \times L$ matrices consisting of these position vectors, i.e., $\mathbf{X} = (\mathbf{x}_1, \cdots \mathbf{x}_M)^T$ and $\mathbf{Y} = (\mathbf{y}_1, \cdots \mathbf{y}_N)^T$. Here $\mathbf{X}^T$ stands for the transposition of $\mathbf{X}$. Hereafter, we assume that nodes in subset V_x are located on the inner circle with radius $r_x = 1$, while nodes in V_y are located on the outer circle with radius $r_y = 2$. Note that $\|\mathbf{x}_m\| = 1$, $\|\mathbf{y}_n\| = 2$.

The centering (Young-Householder transformation) matrices are defined as $\mathbf{H}_M = \mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T$, $\mathbf{H}_N = \mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ where $\mathbf{I}_M$ and $\mathbf{I}_N$ stands for $M \times M$ and $N \times N$ identity matrices, respectively, and $\mathbf{1}_M$ and $\mathbf{1}_N$ are M- and N-dimensional vectors whose elements are all one. By using the double-centered matrix $\mathbf{B} = \{b_{m,n}\}$ that is calculated from the adjacency matrix $\mathbf{A}$ as

$\mathbf{B} = \mathbf{H}_M \mathbf{A} \mathbf{H}_N$, we can consider the following objective function with respect to the position vectors $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_M)^T$ and $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_N)^T$.

$$S(\mathbf{X}, \mathbf{Y}) = \sum_{m=1}^{M} \sum_{n=1}^{N} b_{m,n} \frac{\mathbf{x}_m^T}{r_x} \frac{\mathbf{y}_n}{r_y} + \frac{1}{2} \sum_{m=1}^{M} \lambda_m (r_x^2 - \mathbf{x}_m^T \mathbf{x}_m) + \frac{1}{2} \sum_{n=1}^{N} \mu_n (r_y^2 - \mathbf{y}_n^T \mathbf{y}_n), \tag{5}$$

where $\{\lambda_m \mid m = 1, \cdots, M\}$ and $\{\mu_n \mid n = 1, \cdots, N\}$ correspond to Lagrange multipliers for the spherical constraints, i.e., $\mathbf{x}_m^T \mathbf{x}_m = r_A^2$ and $\mathbf{y}_n^T \mathbf{y}_n = r_B^2$ for $1 \leq m \leq M$ and $1 \leq n \leq N$. By maximizing $S(\mathbf{X}, \mathbf{Y})$ defined in Equation (5), we can obtain our visualization results, $\mathbf{X}$ and $\mathbf{Y}$ for a given bipartite graph.

2.3 Methods for Multigraph

For multigraphs, we denotes the number of links from node u to v, i.e., (u, v), as $m_{u,v}$. Note that favoring or mentioning between a pair of users may occur several times during the observed period. We also denote an in-neighbor node set of node v by $A(v) = \{u; m_{u,v} \neq 0\}$, and an out-neighbor node set of node v by $B(v) = \{w; m_{v,w} \neq 0\}$. Then we can consider a node set $C(k) = \{v; |A(v)| = k\}$ for which the number of in-neighbor nodes is k, and a node set $D(k) = \{v; |B(v)| = k\}$ for which the number of out-neighbor nodes is k. Thus, by using these notations, with respect to the number of neighbors k, we can define the in-neighbor distribution $id(k)$ and the out-neighbor distribution $od(k)$ as follows:

$$in(k) = |C(k)|, \quad on(k) = |D(k)|. \tag{6}$$

Note that in case of simple directed graphs, the in- and out-neighbor distributions are simply called the in- and out-degree distributions, respectively.

Now, we define a set of nodes whose in-degree are not zero by $V^- = \{v \in V; deg^-(v) > 0\}$, and a set of nodes whose out-degree are not zero by $V^+ = \{v \in V; deg^+(v) > 0\}$.

Then, we can define the average in-multiplicity $m^-(v)$ for $v \in V^-$ and the average out-multiplicity $m^+(v)$ for $v \in V^+$ as follow:

$$m^-(v) = \frac{1}{|A(v)|} \sum_{u \in A(v)} m_{u,v}, \quad m^+(v) = \frac{1}{|B(v)|} \sum_{w \in B(v)} m_{v,w}. \tag{7}$$

For a multigraph, we can define the average in-multiplicity m^- and the average out-multiplicity m^+ as follow:

$$m^- = \frac{1}{|V^-|} \sum_{v \in V^-} m^-(v), \quad m^+ = \frac{1}{|V^+|} \sum_{v \in V^+} m^+(v). \tag{8}$$

On the other hand, with respect to number of neighbors $k(> 1)$, we can define the average link multiplicity $im(k)$ for a node set $C(k)$, and the average link multiplicity $om(k)$ for a node set $D(k)$ as follows:

$$im(k) = \frac{1}{|C(k)|} \sum_{v \in C(k)} m^-(v), \quad om(k) = \frac{1}{|D(k)|} \sum_{v \in D(k)} m^+(v). \tag{9}$$

Similarly, for each node $v \in V$, we can define the in-Gini coefficient $g^-(v)$ for $v \in V^-$ and the out-Gini coefficient $g^+(v)$ for $v \in V^+$ as follow:

$$g^-(v) = \frac{\sum_{(u,x)\in A(v)\times A(v)} |m_{u,v} - m_{x,v}|}{2(|A(v)|-1)\sum_{u\in A(v)} m_{u,v}}, g^+(v) = \frac{\sum_{(w,x)\in B(v)\times B(v)} |m_{v,w} - m_{v,x}|}{2(|B(v)|-1)\sum_{w\in B(v)} m_{v,w}}. \tag{10}$$

For a multigraph, we can define the average in-multiplicity m^- and the average out-multiplicity m^+ as follow:

$$g^- = \frac{1}{|V^-|}\sum_{v\in V^-} g^-(v), \quad g^+ = \frac{1}{|V^+|}\sum_{v\in V^+} g^+(v). \tag{11}$$

With respect to number of neighbors $k(> 1)$, we can define the average Gini coefficient $ig(k)$ for a node set $C(k)$, and the average Gini coefficient $og(k)$ for a node set $D(k)$ as follows:

$$ig(k) = \frac{1}{|C(k)|}\sum_{v\in C(k)} g^-(v), \quad og(k) = \frac{1}{|D(k)|}\sum_{v\in D(k)} g^+(v). \tag{12}$$

Here note that the gini coefficient has been widely used for evaluating inequality in a market [7]. We use this index to evaluate inequality between favoring and mentioning.

3 Summary of Data

We briefly explain the data we used in our analysis. These data are retrieved from Favorite, Follow, and Mention of Twitter.

"Favorites" is a function which enables users to bookmark tweets, or to browse them anytime. We constructed a network with the users as nodes, and the Favorer/Favoree relations as links. These data are retrieved from Favotter's "Today's best." [2] during the period from May 1st 2011 to February 12th 2012. Because of Favotter's specification, the retrieved tweets are bookmarked by more than or equal to 5 users. This directed network has 189,717 nodes, 7,077,070 simple links, and 33,456,690 multiple links[3].

"Follow" is the most basic function of Twitter. Users can get the new tweets posted by persons they are interested in by specifying whom to follow. We constructed a network with users who posted more than or equal to 200 tweets as nodes, and the follower/followee [3] relations as links. These data are retrieved from Twitter search [4] as of January 31st 2011. This directed network has 1,088,040 nodes and 157,371,628 simple links. Follow network does not have multiple links because users specify their respective followers only once.

[2] http://favotter.net/

[3] The number of simple links means that we count the multiple links between a pair of nodes as a single link.

[4] http://yats-data.com/yats/

”Mentions” are tweets which has the user’s names of the form ”@Screen_name” in the text. We constructed a network with users as nodes, and send/receive relations as links. These data are retrieved from Toriumi’s data [8] for the period from March 7th 2011 to March 23rd 2011. This directed network has 4,565,085 nodes, 58,514,337 simple links and 193,913,339 multiple links.

Statistics of these networks are described for Tables 1 and 2. Here, WCC1 in Table 1 means the maximal weakly connected components, Em in table 2 means the number of multiple links. Others are defined in section 2.

Table 1: Statistics of simple directed networks

	$\|V\|$	$\|E\|$	$\|V\|_{WCC1}$ ($\|V\|_{WCC1}/\|V\|$)	d	c
Favorites	189,717	7,077,070	189,626 (99.9%)	37.3	0.2109
Follow	1,088,040	157,371,628	1,079,986 (99.3%)	144.6	0.7354
Mentions	4,565,085	58,514,337	1,839,189 (40.3%)	3.2	0.0387

Table 2: statistics of multi directed networks

	$\|V\|$	$\|Em\|$	d	m^-	m^+	g^-	g^+
Favorites	189,717	33,456,690	176.3505	2.1211	1.5024	0.2054	0.0851
Mentions	4,565,085	193,913,339	38.2894	3.6977	3.6574	0.3985	0.2138

Table 1 shows that Mentions network has a smaller WCC1 fraction than the other two networks. This is understandable in view of the communication aspect of Mentions because users do not send @-messages to people whom they do not well. Table 2 shows that Favorites network has smaller m^-, m^+, g^-, and g^+ (see equations 8 and 11) than Mentions. This is understandable because only a few users are heavy favorers and the majorities have much less favorees whereas in Mentions the distribution of the number of mentions of each user is less distorted, which makes the average degree of Mentions network larger than that of Favorites network.

4 Results

In this section, we report the results of analysis using various measures explained in 2.

4.1 Simple Directed Graph

As seen from Table 1, Favorites and Follow networks have each a large weakly connected component which includes almost all nodes but Mentions network is not so. Since Mentions network is too large to analyze for all nodes, we use WCC1 in the following analysis for Mentions network.

Degree Distribution Figures 2, 3, 4, 5, 6, and 7 are the results of degree distribution of the three networks. Blue and red diamond marks indicate *id* and *od* (see equation (3)), respectively. The vertical axis indicates the number of nodes in logarithmic scale. From these pictures, we see that all the networks can be said to have a scale-free property for both in-degree or out-degree.

Network Motif Figures 8 and 9 are the results of network motif analysis. The horizontal axis indicates the motif number explained in 4. In Figure 8 the vertical axis indicates the frequency of appearance in logarithmic scale, and in Figure 9 the vertical axis indicates z-score (see equation (4)) in logarithmic scale. Magenta and cyan bars mean positive score and negative score respectively. From these figures, we see that there are three predominant motifs: patterns 13, 12, and 8, which are all characterized by having mutual links, The results of Follow and Mentions networks are similar to these figures, so we omit showing these results.

4.2 Visualization of Bipartite Graph

Figure 10 is the result of visualization of bipartite graph of Favorites. In this analysis we used the data retrieved from only July 1st to 7th 2011 because so many links obscure the graph. Nodes on the outer circle are Favorers, and nodes on the inner circle are Favorees. Blue and Red nodes are users who are ranked Favorer/Favoree's top 10. Only links with more than or equal to 10 multiplicity are shown by gray lines.

NHK_PR is the official account of NHK's PR section[5], and sasakitoshinao is the account of freelance journalist. His tweets are on serious and important topics, for instance, current news or opinions about it. On the other hand, kaiten_keiku and Satomii_Opera are regular users of Twitter, and their tweets are often negative and/or "geeky".

From this figure, we see there are two clusters of popular users which are characterized by their content of tweets, one with serious and important tweets and the other with negative and/or geeky tweets.

4.3 Multiple Directed Graph

In this subsection, we show the results of analysis using the measures explained in 2.3. In all the figures below (Figures 11 to 22), plots in blue squares are for

[5] Japan Broadcasting Corporation

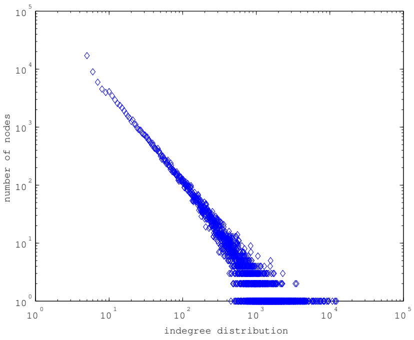


Fig. 2: Favorites network in-degree

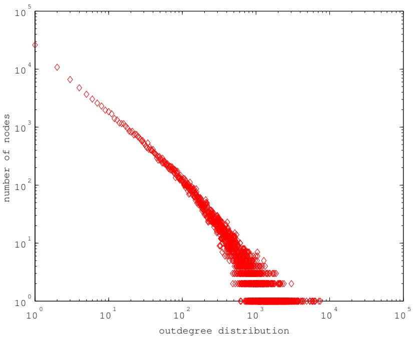


Fig. 3: Favorites network out-degree

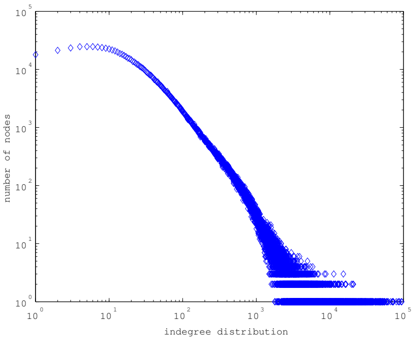


Fig. 4: Follow network in-degree

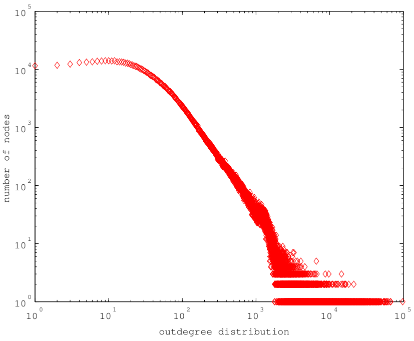


Fig. 5: Follow network out-degree

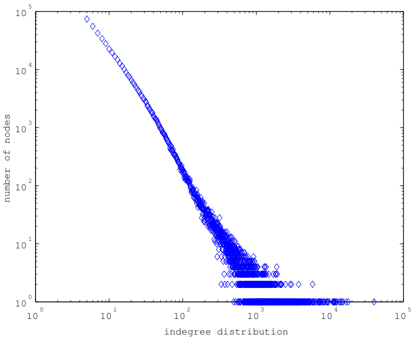


Fig. 6: Mentions network in-degree

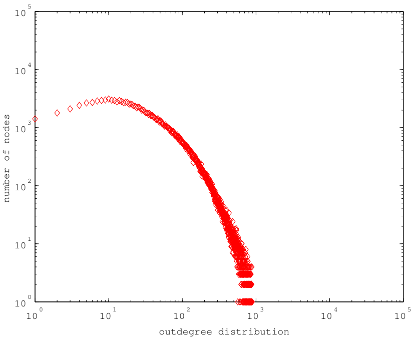


Fig. 7: Mentions network out-degree

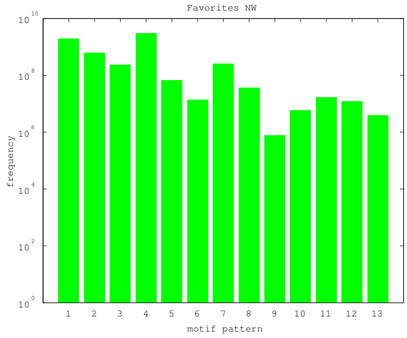


Fig. 8: Favorites network motif (frequency)

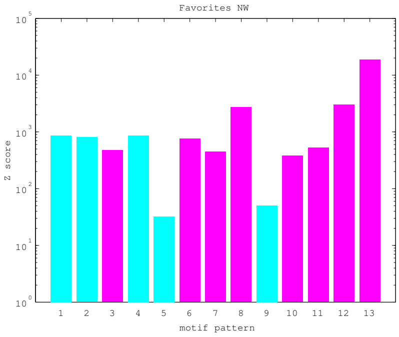


Fig. 9: Favorites network motif (z-score)

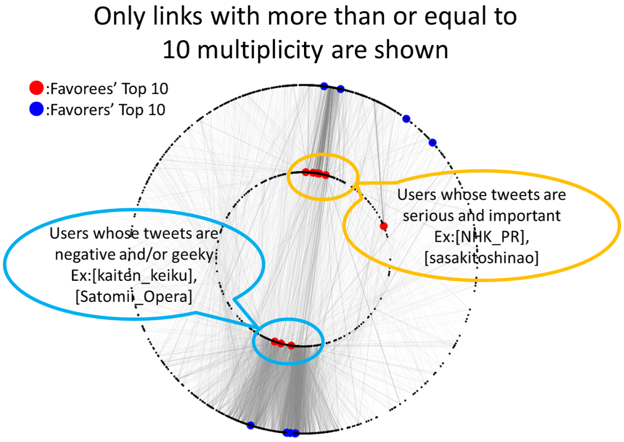

Only links with more than or equal to
10 multiplicity are shown
:Favorees' Top 10
:Favorers' Top 10
Users whose tweets are serious and important Ex:[NHK_PR], [sasakitoshinao]
Users whose tweets are negative and/or geeky Ex:[kaiten_keiku], [Satomii_Opera]

Fig. 10: Bipartite Graph Visualization

in-degree, plots in red squares are for out-degree and plots in green circles are for randomized networks. Horizontal axes are all in logarithmic scale.

Degree Distribution Figures 11, 12, 13 and 14 are the results of degree distribution (see equation (6)) for Favorites and Mentions networks. The vertical axes are frequency (the number of nodes) in logarithmic scale.

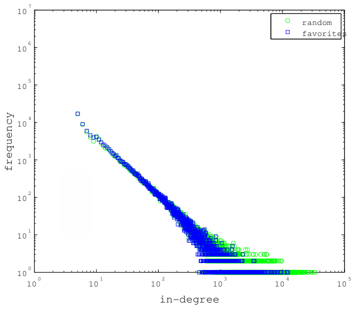


Fig. 11: Favorites in-degree

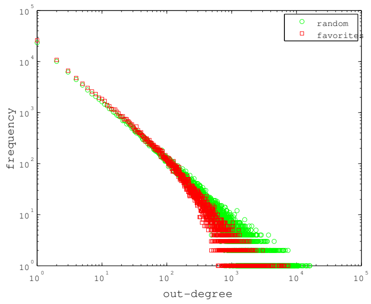


Fig. 12: Favorites out-degree

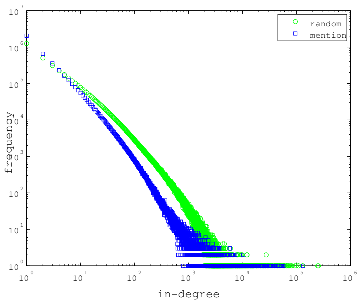


Fig. 13: Mentions in-degree

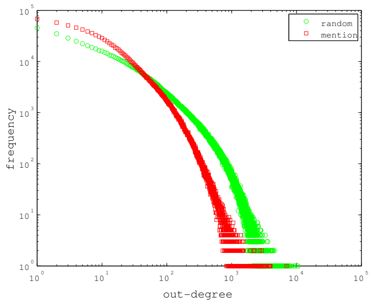


Fig. 14: Mentions out-degree

From these figures, we see that both networks have a scale-free property, same as the simple directed networks 4.1. We notice that the distributions for the randomized Mentions network are shifted right to the real Mentions network, but this is not so for Favorites network.

Average Multiplicity Figures 15, 16, 17 and 18 are the average multiplicity (see equation (7)) for the both networks. The vertical axes are in logarithmic scale.

We notice the difference in correlation between the two networks. On the average, there are positive correlations between the average multiplicity and the degree for Favorites network (Figures 15 and 16), but the correlations change from positive to negative as the degree increases for Mentions network (Figures

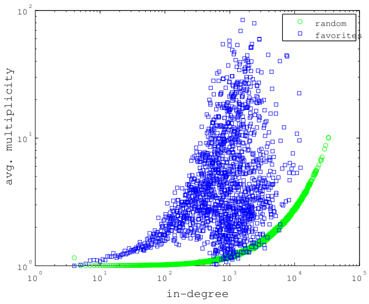


Fig. 15: Favorites in-degree

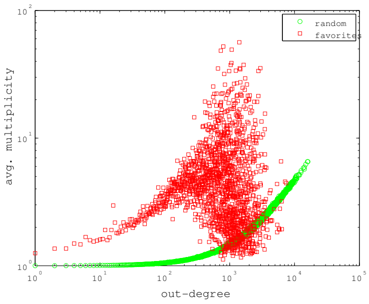


Fig. 16: Favorites out-degree

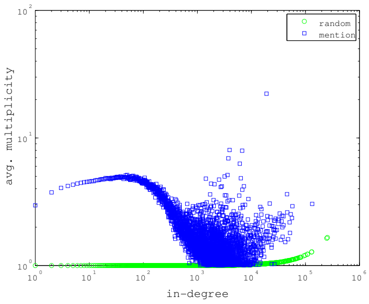


Fig. 17: Mentions in-degree

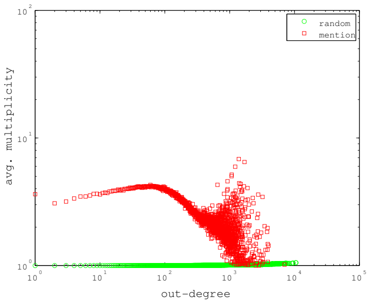


Fig. 18: Mentions out-degree

17 and 18). Furthermore, the average multiplicity of randomized Favorites network behaves similarly to the real Favorites network, but that of randomized Mentions network is almost flat across all the range of degree.

Gini coefficient Figures 19, 20, 21 and 22 are the results of Gini coefficient (see equation (10) for the both networks. The vertical axes are in linear scale.

Correlations between the Gini coefficient and the degree and the relation between the real and the randomized networks are similar to those for the average multiplicity, i.e., positive correlations for Favorites network (Figures 19 and 20), positive to negative correlations for Mentions network (Figures 21 and 22) and more positive correlations for the randomized Favorites network than the randomized Mentions network.

5 Discussion

The results in subsections 4.1 and 4.3 revealed that all the three networks have the scale-free property, but we notice that the variance in the degree distributions for Mentions network is smaller in high out-degree nodes than others. We conjecture that this is due to the communication aspect of Mention function, i.e. users do not send many @-messages to people they do not know well and, thus,

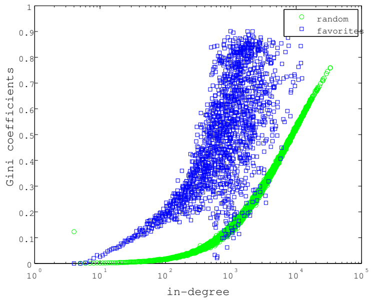


Fig. 19: Favorites in-degree

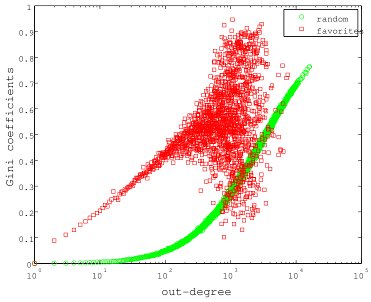


Fig. 20: Favorites out-degree

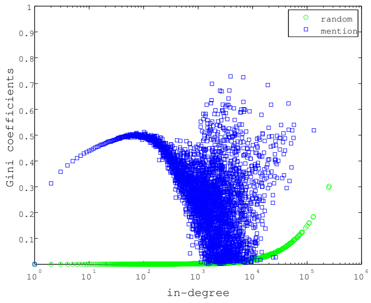


Fig. 21: Mentions in-degree

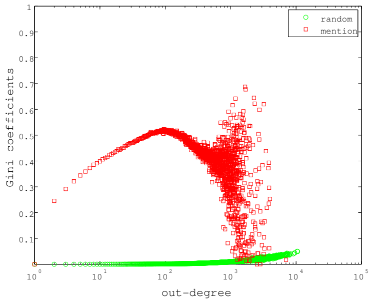


Fig. 22: Mentions out-degree

there are probably no big hub nodes in Mentions network. Further, this also explains that the fraction of the maximal weakly connected component (defined in subsection 3) is smaller than the other networks.

The results in subsection 4.1 revealed that there are a few numbers of predominant motifs that are characteristic of having mutual links. This accounts for the fact that, taking Favorites as example, mutual links are easily created between users who have similar tastes because Favorites network is driven by preference.

The results in subsection 4.2 that there are two clusters of popular users each corresponding to a particular type of tweets are quite natural and understandable. Whether these two are the unique tweets and there are no other such tweets remains to be explored.

The results in subsection 4.3 indicate that there are substantial difference in the distributions of multiplicity and Gini coefficient for high degree nodes between Favorites and Mentions networks. This is explainable considering the difference in nature of the two functions, Mentions network is driven by communications between users. Sending/receiving of @-message to/from many people become less practical, thus less frequent for high degree nodes. Favorites network is driven by preference. Expressing preference (bookmarking Favorees' tweets) is much easier than sending/receiving message, thus relatively more frequent for high degree nodes.

The results in subsection 4.3 revealed that there are positive correlations between the Gini coefficient and the degree for all the range of degree for Favorites network, but not so for Mentions network. This may suggest that Favorers in high out-degree tends to preferentially bookmark specific Favorees' tweets, and vice versa for Favorees in high in-degree.

6 Conclusion

With the final goal of constructing a new user preference model in daily activities in mind, we analyzed, from the network structure perspective, the similarity and difference in the user behavior of the three functions of Twitter: Favorite, Follow and Mention. User behavior is embedded in the logs that users carried out these functions, which are represented by directed graphs. Favorite function was analyzed using three different graph representations: a simple graph, a multigraph and a bipartite graph, Follow function by one graph representation: a simple graph, and Mention function by two graph representations: a simple graph and a multigraph. We used three real world Twitter logs to create these directed graphs and performed various kinds of analysis using several representative measures for characterizing structural properties of graphs, and obtained several salient features.

Major findings are that 1) Favorites and Followers networks have a very large connected component but Mentions network is not, 2) all the three networks (both simple and multiple) have the scale-free property in degree distribution, 3) all three networks (simple) have predominant three-node motifs having mutual links, 4) Favorites networks have freaks in Gini coefficient (one of the measures), 5) Favorites networks have two clusters of popular users, and 6) Favorites and Mentions networks differ in structure for high degree nodes in case of multigraph representation reflecting that Favorite operation is much easier than Mention operation although they are similar in case of simple graph representation.

As an immediate future work, we plan to obtain betweenness centrality, closeness centrality, or k-core percolation of Favorites network represented as a multigraph to further characterize use behavior and hopefully to extract enough regularity to model user preference, and pursue the literature review and usefulness of the model.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-11-4111, JSPS Grant-in-Aid for Scientific Research (C) (No. 22500133).

References

1. U. Brandes and T. Erlebach (Eds.), "Network analysis", LNCS 3418, pp. 293-317, Springer-Verlag, 2005.

2. T. Fushimi, Y. Kubota, K. Saito, M. Kimura, H. Motoda, and K. Ohara, "Speeding up bipartite graph visualization method", Proc. of the 24th Australasian Joint Conference on Artificial Intelligence (AI2011).
3. B.A. Huberman, D.M. Romero and F. Wu, "Social networks that matter: Twitter under the microscope", First Monday, Vol. 14, No. 1, 2009.
4. H.Kwak, C.Lee, H.Park, and S.Moon, "What is Twitter, a social network or a news media?", In Proceedings of the 19th international conference on World Wide Web, pp.591-600, 2010.
5. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii and U. Alon, "Network motifs: simple building blocks of complex networks", Science 298, pp. 824–827, 2002.
6. M. E. J. Newman,"The structure and function of complex networks", SIAM Review, Vol.45, pp.167–256, 2003.
7. M.J. Salganik, P.S. Dodds, and D.J. Watts. "Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market", Science 311, pp. 854–856, 2006.
8. F. Toriumi, K. Shinoda, S. Kurihara, T. Sakaki, K. Kazama, I. Noda,"Disaster Changes Social Media", In Proceedings of the 7th Conference of JWEIN, 2011.

Extracting Communities in Networks based on Functional Properties of Nodes

Takayasu Fushimi[1], Kazumi Saito[1], and Kazuhiro Kazama[2]

[1] Graduate School of Management and Information of Innovation,
University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
{j11507,k-saito}@u-shizuoka-ken.ac.jp
[2] Nippon Telegraph and Telephone Corporation, Network Innovation Laboratories
3-9-11 Midori-cho, Musashino-shi, Tokyo 180-8585,
kazama@ingrid.org

Abstract. We address the problem of extracting the groups of functionally similar nodes from a network. As functional properties of nodes, we focus on hierarchical levels, relative locations and/or roles with respect to the other nodes. For this problem, we propose a novel method for extracting functional communities from a given network. In our experiments using several types of synthetic and real networks, we evaluate the characteristics of functional communities extracted by our proposed method. From our experimental results, we confirmed that our method can extract functional communities, each of which consists of nodes with functionally similar properties, and these communities are substantially different from those obtained by the Newman clustering method.

1 Introduction

Finding groups of functionally similar nodes in a social or information network can be a quite important research topic in various fields ranging from computer science to sociology. Hereafter, such a node group is simply referred to as a functional community. In fact, each node which typically corresponds to a person in a social network may have a wide variety of functional properties such as status, ranks, roles, and so forth, as described in [1]. However, conventional methods for extracting communities as densely connected subnetworks, which include the Newman clustering method based on a modularity measure [2], and normalized cut [3] or ratio cut [4] method based on the spectral graph analysis, cannot directly deal with such functional properties. Evidently, conventional notions of densely connected subnetworks such as k-core [5], k-dense [6] and k-clique [7] cannot work for this purpose. Namely, it is naturally anticipated that these existing methods have an intrinsic limitation for extracting functional communities.

In this study, as typical functional properties of nodes, we especially focus on hierarchical levels, relative locations and/or roles with respect to the other nodes. This implies that there exist some functionally similar nodes even if they are not directly connected with each other. For instance, in case of a network of employees relationships in a company, we can naturally assume it to have a hierarchical property, where the top node corresponds to the president, and in turn, the successive levels of nodes correspond to managers, section leaders, and so on. For example, our objective might be to extract a group of section leaders as a functional community in the network, even though they may not have direct connections with each other. Similarly, even in case of a hyperlink network of Web pages in a site, we can also assume it to have a hierarchical property, where the top node corresponds to the top page at this site served as an entrance, and in turn, the successive levels of nodes may correspond to Web pages containing more specific topics. Then our objective might be to extract a group of Web pages with the same level of topic specificity. Here we should emphasize that extracting these types of communities can be a quite tough problem for the conventional community extraction methods because these existing methods mainly focus on link densities among each subnetwork and between subnetworks.

In this paper, we propose a novel method for extracting functional communities from a given network. This algorithm consists of two steps: the method first assigns a feature vector to each node, which is assumed to be some functional properties, by using calculation steps of PageRank scores [8] for nodes from an initial score vector. Then, in a case that the supposed number of functional communities is K, the method divides all the node into K groups by using the K-medians clustering method based on the cosine similarity between a pair of the feature vectors. In our experiments using several types of synthetic and real networks, we evaluate the characteristics of functional communities extracted by our proposed method. To this end, we utilize the visualization result of each network where each functional community is indicated by a different color marker, and these results are contrasted to those obtained by the Newman clustering method [2].

This paper is organized as follows: after explaining two component algorithms in Section 2, we describe a detail of our proposed method in Section 3. Then, by using a number of visualized networks, in comparison to standard communities extracted by the Newman clustering method, we qualitatively evaluate the characteristics of the extracted functional communities in Section 4. Finally, we describe our conclusion in Section 5.

2 Component Algorithms

In this section, for the sake of convenience, we explain two existing methods, PageRank [8] and K-medians. These are used as component algorithms for our newly proposing method.

2.1 PageRank Revisited

For a given Web hyperlink network (directed graph), we identify each node with a unique integer from 1 to $|V|$. Then we can define the adjacency matrix

$\boldsymbol{A} \in \{0,1\}^{|V|\times|V|}$ by setting $a(u,v) = 1$ if $(u,v) \in E$; otherwise $a(u,v) = 0$. A node can be self-looped, in which case $a(u,u) = 1$. For each node $v \in V$, let $F(v)$ and $B(v)$ denote the set of child nodes of v and the set of parent nodes of v, respectively, $F(v) = \{w \in V;\ (v,w) \in E\}$, $B(v) = \{u \in V;\ (u,v) \in E\}$. Note that $v \in F(v)$ and $v \in B(v)$ for node v with a self-loop.

Then we can consider the row-stochastic transition matrix $\boldsymbol{P}$, each element of which is defined by $p(u,v) = a(u,v)/|F(u)|$ if $|F(u)| > 0$; otherwise $p(u,v) = z(v)$, where $\boldsymbol{z}$ is some probability distribution over nodes, i.e., $z(v) \geq 0$ and $\sum_{v\in V} z(v) = 1$. This model means that from dangling Web pages without outlinks ($F(u) = \emptyset$), a random surfer jumps to page v with probability $z(v)$. The vector $\boldsymbol{z}$ is referred to as a personalized vector because we can define $\boldsymbol{z}$ according to user's preference.

Let $\boldsymbol{y}$ denote a vector representing PageRank scores over nodes, where $y(v) \geq 0$ and $\sum_{v\in V} y(v) = 1$. Then using an iteration-step parameter s, PageRank vector $\boldsymbol{y}$ is defined as a limiting solution of the following iterative process,

$$\boldsymbol{y}_s^T = \boldsymbol{y}_{s-1}^T \left((1-\alpha)\boldsymbol{P} + \alpha \boldsymbol{e}\boldsymbol{z}^T\right) = (1-\alpha)\boldsymbol{y}_{s-1}^T\boldsymbol{P} + \alpha\boldsymbol{z}^T, \tag{1}$$

where $\boldsymbol{a}^T$ stands for a transposed vector of $\boldsymbol{a}$ and $\boldsymbol{e} = (1,\cdots,1)^T$. In the Equation (1), α is referred to as the uniform jump probability. This model means that with the probability α, a random surfer also jumps to some page according to the probability distribution $\boldsymbol{z}$. The matrix $((1-\alpha)\boldsymbol{P} + \alpha\boldsymbol{e}\boldsymbol{z}^T)$ is referred to as a Google matrix. The standard PageRank method calculates its solution by directly iterating Equation (1), after initializing $\boldsymbol{y}_0$ adequately. One measure to evaluate its convergence is defined by

$$\|\boldsymbol{y}_s - \boldsymbol{y}_{s-1}\|_{L1} \equiv \sum_{v\in V} |y_s(v) - y_{s-1}(v)|. \tag{2}$$

Note that any initial vector $\boldsymbol{y}_0$ can give almost the same PageRank scores if it makes Equation (2) almost zero because the unique solution of Equation (1) is guaranteed.

2.2 *K*-medians Revisited

For a given set of objects (or nodes), denoted by $V = \{v, w, \cdots\}$, the K-medians method first selects K representative objects $\mathcal{R} \subset V$ according to the following objective function to be maximized.

$$f(\mathcal{R}) = \sum_{v\in V} \max_{r\in\mathcal{R}}, \rho(v,r). \tag{3}$$

Here $\rho(v,r)$ stands for a similarity measure between a pair of objects, v and r. Then, from the obtained K representative objects $\mathcal{R} = \{r_1,\cdots,r_K\}$, the method determines the K clusters, $\{\mathcal{C}_1,\cdots,\mathcal{C}_K\}$, by using the following formula.

$$\mathcal{C}_k = \{v \in V; r_k = \arg\max_{r\in\mathcal{R}}, \rho(v,r)\}. \tag{4}$$

Finally, the method outputs $\{\mathcal{C}_1, \cdots, \mathcal{C}_K\}$ as the result.

In order to maximize Equation (3) with respect to $\mathcal{R}$, due to simplicity we employ a greedy algorithm shown below.

1. Initialize $k \leftarrow 1$ and $\mathcal{R} \leftarrow \emptyset$;
2. Select $r_k = \arg\max_{w \in V \setminus \mathcal{R}} \{f(\mathcal{R})\}$, and set $\mathcal{R} \leftarrow \mathcal{R} \cup \{r_k\}$;
3. If $k = K$, output $\mathcal{R} = \{r_1, \cdots, r_K\}$ and terminate ;
4. Set $k \leftarrow k + 1$ and return to step 2;

Here note that in virtue of the submodularity of the objective function defined in Equation (3), we can obtain a unique greedy solution whose worst case quality is guaranteed [9].

3 Proposed Method

In this section, we describe our proposed method for extracting functional communities. Our method utilizes the PageRank score vectors at each iteration step s, i.e., $\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_S\}$. Here, S stands for the final step when the PageRank algorithm converges. Then, for each node $v \in V$, we can consider an S-dimensional vector defined by $\boldsymbol{x}_v = (y_1(v), \cdots, y_S(v))^T$, where $y_s(v)$ means the PageRank score of node v at iteration step s. In our method, $\boldsymbol{x}_v$ is regarded as a functional property vector of node v.

Here we note a reason why we employ the vector described above. Basically, we assume that functional properties of nodes, such as hierarchical levels, relative locations and/or roles with respect to the other nodes are embedded into the network structure. On the other hand, the PageRank scores at each iteration step also reflect the network structure. Therefore, as an approximation, we can consider that functional properties are also represented by the vector $\boldsymbol{x}_v$.

In order to divide all nodes into the K groups, our method employs the K-medians algorithm described in the previous section. To this end, we need to define an adequate similarity $\rho(u, v)$ between the nodes u and v. In our proposed method, for each pair of functional property vectors, we employ the following cosine similarity.

$$\rho(u, v) = \frac{\boldsymbol{x}_u^T}{||\boldsymbol{x}_u||} \frac{\boldsymbol{x}_v}{||\boldsymbol{x}_v||}, \tag{5}$$

where $||\boldsymbol{x}_v||$ stands for the standard L2 norm.

For a given network $G = (V, E)$ and the number K of functional communities, we summarize our proposed algorithm below.

1. Calculate the PageRank score vectors at each time step $\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_S\}$;
2. Construct the functional property vector $\mathbf{x}_v$ for each node $v \in V$;
3. Calculate the cosine similarity $\rho(u, v)$ of $\mathbf{x}_u$ and $\mathbf{x}_v$ for all node pair;
4. Divide all nodes into K clusters according to the similarity $\rho(u, v)$ by the K-medians method;
5. Output functional communities $\{\mathcal{C}_1, \cdots, \mathcal{C}_K\}$;

4 Experimental Evaluation

In this section, using several types of synthetic and real networks, we experimentally evaluate the characteristics of functional communities extracted by our proposed method. For this purpose, we utilize the visualization result of each network where each functional community is indicated by a different color marker, and these results are contrasted to those obtained by the Newman method [2].

4.1 Network Data

We describe a detail of four networks used in our experiments.

First one is a synthetic network with a hierarchical property, just like an employee relationships or Web hyperlinks network. In this hierarchical network, we can assume two types of nodes, central (or high status) and peripheral (or low status) nodes. As shown later in Fig. 1, in terms of its basic network statistics, the central nodes are characterized by relatively high degree and low clustering coefficients, while the peripheral nodes by relatively low degree and high clustering coefficients. We generated this network according to Ravasz et.al. [10]. Hereafter, this network is referred to as Hierarchical network.

Second one is a two dimensional-grid network implemented as a set of 10 × 10 lattice points. Evidently, as shown later in Fig. 2, because of the regular structure, dividing this network into several portions does not make sense in the aspects of standard community extraction. Whereas, we can consider a functional property in terms of relative locations to other nodes, i.e., the relative closeness to the center position. Hereafter, this network is referred to as Lattice network.

Third one is a social network of people belonging to a karate circle, which has been widely used as a benchmark network. As shown later in Fig. 3, we see a number of hub nodes, which play an important role to connect other nodes. Namely, we can assume that some group of nodes has a similar role with respect to the other nodes. Hereafter, this network is referred to as Karate network [11].

Forth one is a hyperlink network of a Japanese university Web site, where we obtained this network by crawling the Web site as of Aug. 2010. As shown later in Fig. 4, there exist a number of unique characteristics in this network. Namely, we can assume that some group of Web pages has a similar topic specificity level. Hereafter, this network is referred to as Hosei network [2].

Table 1 shows the basic statistics of the Hierarchical, Lattice, Karate and Hosei networks. Here, C and L denote the averages of clustering coefficients and shortest path lengths, respectively.

4.2 Experimental Settings

We first explain the settings of our proposed algorithm. In order to calculate the PageRank score vectors, we set the initialized vector to $\boldsymbol{y}_0 = (1/|V|, \ldots, 1/|V|)^T$,

[2] The site name and its address are "Faculty of Computer and Information Sciences, Hosei University" and http://cis.k.hosei.ac.jp/ , respectively.

Table 1. Basic statistics of networks.

network	$\|V\|$	$\|E\|$	C	L
Hierarchical	125	410	0.84	2.13
Lattice	100	180	0.00	4.59
Karete	34	78	0.57	2.03
Hosei	600	1299	0.54	4.22

and the convergence criterion defined in Equation 2 is implemented as $||\boldsymbol{y}_s - \boldsymbol{y}_{s-1}||_{L1} < 10^{-12}$. The number K of communities to be extracted is changed from $K = 2$ to 10.

As mentioned earlier, we attempt to clarify the characteristics of the functional communities extracted by our method, in comparison to standard communities extracted by the Newman clustering method [2]. Hereafter, such a standard community is simply referred to as a Newman community. The Newman method is basically designed to obtain densely connected subnetworks by maximizing a modularity measure.

Finally, we describe methods to visualize each network. In Hierarchical network, we employ nodes' positions as displayed by Ravasz et.al. [10]. As for Lattice network, we can regularly assign the positions to nodes. In cases of Karate and Hosei networks, the cross-entropy embedding method [12] is used to determine the positions of nodes.

4.3 Experimental results

We show the experimental results of Hierarchical network at $K = 5$ in Fig. 1. Here note that this network consists of five portions of densely connected subnetworks, as observed in Fig. 1. Thus, as an example, we selected this number, $K = 5$. As expected, from Fig. 1(a), we see that our method could extract reasonable functional communities, each of which consists of nodes with the similar hierarchical levels, just like employees with same position such as the president, managers, or general staffs. On the other hand, from Fig. 1(b), we see that the Newman method extracted standard communities, each of which is characterized as a densely connected subnetwork, just like employees belonging to the same department or section.

We show the experimental results of Lattice network at $K = 3$ in Fig. 2. Here recall that in the aspects of standard community extraction, dividing this network into several portions does not make sense. Thus, as an example, we selected this relatively small number, $K = 3$. From Fig. 2(a), we see that our method could extract reasonable functional communities, each of which consists of nodes with the similar relative locations, i.e., the relative closeness to the center position. On the other hand, as shown in Fig. 2(b), we can hardly make sense to the communities extracted by the Newman method.

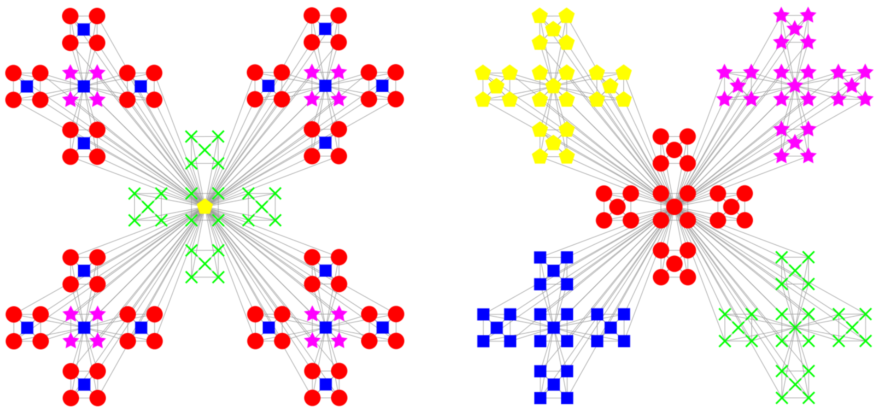
We show the experimental results of Karate network at $K = 2$ in Fig. 3. Here note that this network consists of two portions of densely connected subnetworks, as observed in Fig. 3. Thus, as an example, we selected this number, $K = 2$. From Fig. 3(a), we see that our method could extract reasonable functional communities, each of which consists of nodes with different roles with respect to the other nodes, i.e., groups of hub nodes and the other nodes. On the other hand, from Fig. 3(b), we see that the Newman method extracted standard communities, each of which is characterized as a densely connected subnetwork.

We show the experimental results of Hosei network at $K = 10$ in Fig. 4. Here note that this network consists of several portions of characteristically connected subnetworks, as observed in Fig. 4. Thus, as an example, we selected this relatively large number, $K = 10$. From Fig. 4(a), we see that our method extracted several communities, each of which consists of nodes with similar connection patterns. In order to more closely investigate these extracted communities, we focused on a particular community indicated by small blue squares surrounding with large transparent squares in Fig. 4(a). From our examination of these Web pages belonging to this community, we realized that these Web pages correspond to annual reports of each year produced by faculty members. Namely, it is assumed that these Web pages in this community have a similar topic specificity level. Thus, we can consider that our method could extract a piece of reasonable functional communities in the sense described above. On the other hand, from Fig. 4(b), we see that the Newman method divided the functional community focused above into several communities.

From our experimental results using these networks with different characteristics, we confirmed that our method could extract functional communities, each of which consists of nodes with similar functional properties such as hierarchical levels, relative locations and/or roles with respect to the other nodes. These results indicate that our method is promising for tasks of extracting functional communities with these properties. On the other hand, the Newman method extracted standard communities characterized by densely connected subnetworks. From these results, we see that these functional communities extracted by our method are substantially different from those obtained by the Newman method.

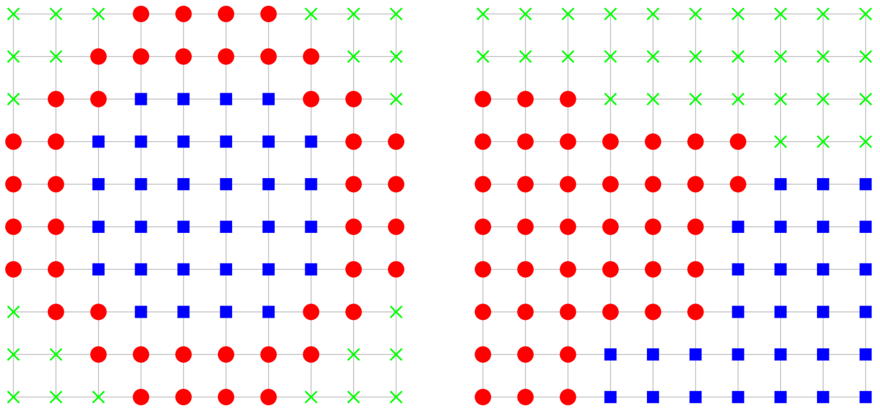
5 Conclusion

We addressed the problem of extracting the groups of functionally similar nodes from a network. In this paper, such a node group was simply referred to as a functional community. As functional properties of nodes, we focused on hierarchical levels, relative locations and/or roles with respect to the other nodes, and proposed a novel method for extracting functional communities from a given network. In our experiments using several types of synthetic and real networks, we evaluated the characteristics of functional communities extracted by our proposed method. From our experimental results, we confirmed that our method could extract functional communities, each of which consists of nodes with functionally similar properties, and these communities were substantially different from those obtained by the Newman clustering method. In future, we plan to evaluate our method using various networks.



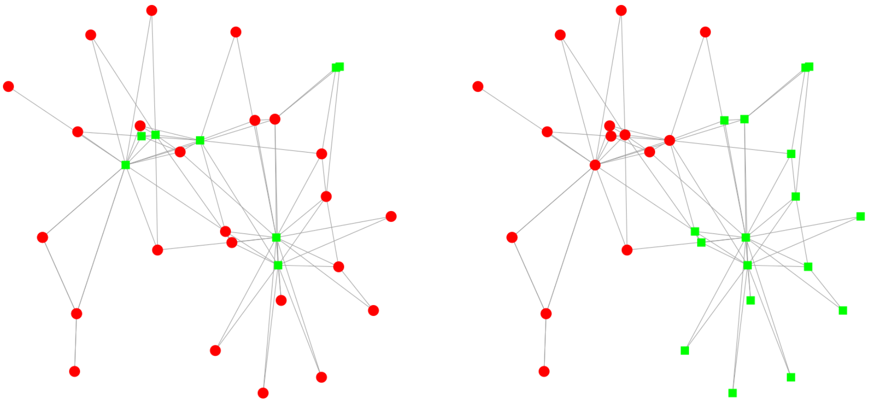
(a) Functional community (b) Newman community

Fig. 1. Hierarchical Network ($K = 5$)



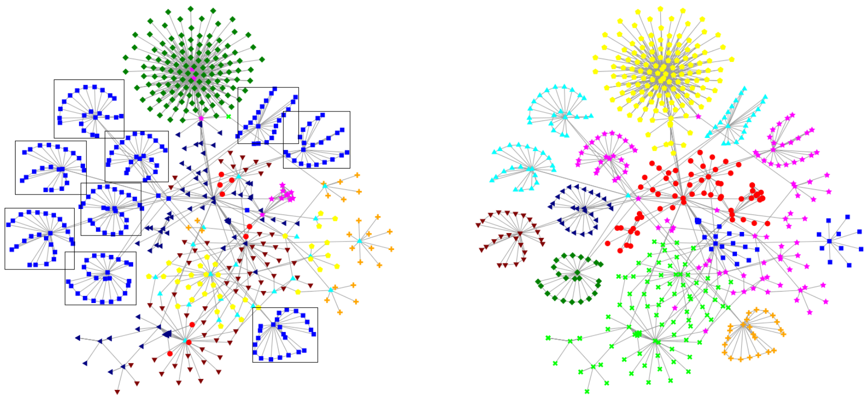
(a) Functional community (b) Newman community

Fig. 2. Lattice Network ($K = 3$)



(a) Functional community (b) Newman community

Fig. 3. Karate Network ($K = 2$)



(a) Functional community (b) Newman community

Fig. 4. Hosei Network ($K = 10$)

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-11-4111, JSPS Grant-in-Aid for Scientific Research (C) (No. 23500128), and NTT Network Innovation Laboratories.

References

1. M. E. J. Newman. Mixing patterns in networks. *Physical Review E*, Vol. 67, No. 2, pp. 026126, 2003.
2. M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, Vol. 69, No. 6, pp. 066133, 2004.
3. J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 888–905, 2000.
4. L. Hagen and A. B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 11, No. 9, pp. 1074–1085, 1992.
5. S. B. Seidman. Network structure and minimum degree. *Social Networks*, Vol. 5, No. 3, pp. 269 – 287, 1983.
6. K. Saito, T. Yamada, and K. Kazama. The k-dense method to extract communities from complex networks. In Djamel Zighed, Shusaku Tsumoto, Zbigniew Ras, and Hakim Hacid, editors, *Mining Complex Data*, Vol. 165 of *Studies in Computational Intelligence*, pp. 243–257. Springer Berlin / Heidelberg, 2009.
7. G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, Vol. 435, pp. 814–818, 2005.
8. A. N. Langville and C. D. Meyer. Deeper inside pagerank. *Internet Mathematics*, Vol. 1, Vol. 3, pp. 335 – 380, 2004.
9. G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, Vol. 14, pp. 265–294, 1978.
10. E. Ravasz and A. L. Barabási. Hierarchical organization in complex networks. *Physical Review E*, Vol. 67, No. 2, pp. 026112, 2003.
11. W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, Vol. 33, pp. 452–473, 1977.
12. T. Yamada, K. Saito, and N. Ueda. Cross-entropy directed embedding of network data. In *Proceedings of the 20th International Conference on Machine Learning (ICML03)*, pp. 832–839, 2003.

Opinion Formation by Voter Model with Temporal Decay Dynamics

Masahiro Kimura[1], Kazumi Saito[2], Kouzou Ohara[3], and Hiroshi Motoda[4]

[1] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[2] School of Administration and Informatics, University of Shizuoka
Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 252-5258, Japan
ohara@it.aoyama.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. Social networks play an important role for spreading information and forming opinions. A variety of voter models have been defined that help analyze how people make decisions based on their neighbors' decisions. In these studies, common practice has been to use the latest decisions in opinion formation process. However, people may decide their opinions by taking account not only of their neighbors' latest opinions, but also of their neighbors' past opinions. To incorporate this effect, we enhance the original voter model and define the temporal decay voter (TDV) model incorporating a temporary decay function with parameters, and propose an efficient method of learning these parameters from the observed opinion diffusion data. We further propose an efficient method of selecting the most appropriate decay function from among the candidate functions each with the optimized parameter values. We adopt three functions as the typical candidates: the exponential decay, the power-law decay, and no decay, and evaluate the proposed method (parameter learning and model selection) through extensive experiments. We, first, experimentally demonstrate, by using synthetic data, the effectiveness of the proposed method, and then we analyze the real opinion diffusion data from a Japanese word-of-mouth communication site for cosmetics using three decay functions above, and show that most opinions conform to the TDV model of the power-law decay function.

1 Introduction

Social networking services (SNSs) on the Internet, such as Facebook, Twitter and Digg, have become so popular and use of these services is now a part of our daily activities. Large networks formed by these services play an important role as a medium for spreading diverse information including news, ideas, opinions, and rumors [18, 17, 8, 6]. Users of these services can share their interests or opinions to each other. The resulting social networks and the information propagated therein have great influence on and drastically change our decision making processes and behaviors in daily life. Thus, many attempts have been made to investigate the spread of influence in social networks [15, 5, 21].

One such typical and well studied problem in social network analysis is the *influence maximization problem*, which is finding a limited number of influential nodes that are effective for spreading information [10, 11, 16, 3, 4]. What is common to these studies is that models used allow a node in the network to take only one of the two states, i.e., either active or inactive, because the focus is on *influence*. However, we need a model in which a node can take multiple states for such applications in which a user can choose one from multiple choices. For example, a mobile phone user may change his/her current carrier to the one which the majority of his/her neighbors are using. To model this kind of opinion formation dynamics, a node in the network has to be able to take one of many possible choices as its state. A *voter model* would be the one which is most suitable for this purpose. It is one of the most basic stochastic process models, where a node decision is influenced by its neighbors' decisions [20, 9, 7, 2, 22]. We proposed two variants of voter model in our past work: the *value-weighted voter model* that considers opinion values [12], and the *value-weighted mixture voter model* that, in addition to the opinion values, considers the effect of anti-majoritarians, i.e., those people who do not agree with the majority and support the minority opinion [13].

In this paper we also address the problem of opinion formation on the social network, but we especially focus on the fact that our decision may be influenced not only by our neighbors' and our own latest opinions, but also by the neighbors' and our own past opinions. For example, assume that you and your friends have long supported a certain political party, but many of your friends have started changing their supporting party to a different one very recently. Under this situation, you may still stick to your opinion and keep supporting the party, or you may change your mind and follow your neighbors' opinions. This means that your current opinions are influenced not only by the neighbors' latest opinions but also by their past opinions including your own opinions. It is, thus, important to consider all the past opinions in making the current decision. Nonetheless, all the voter models including the two variants mentioned above consider only the latest opinions of its neighbors including itself when updating the opinion of a node.

With this in mind we enhance the original voter model and define the *temporal decay voter (TDV) model* that takes into account all the past opinions discounting the effect of older opinions by using a temporal decay function. The work most closely related to our approach would be the work by Koren [14] which is in the context of recommender systems, where several time drifting user preference models are proposed, some of which adopt a temporal decay function that discounts the effect of older ratings to items. The approach in Koren's work is, unlike our approach, cannot utilize all the past ratings given by a user for an identical item because the user-item matrix that they use does not allow multiple ratings to be stored. In addition, due to the framework of collaborative filtering, it requires the rating history involving multiple items, while our approach can model the temporal dynamics of opinions for a single item. Thus, it does not make sense to compare Koren's approach with ours.

Our major contribution is the following four: 1) the TDV model, 2) an algorithm of learning the parameters of the temporal decay function from the observed opinion spreading data, 3) a model selection method that determines the most appropriate decay function for given data, and 4) new finding regarding the decay model from the analysis of the real data. The model parameters are learned by an efficient iterative algorithm which maximizes the likelihood function. Three representative decay functions are employed, although the framework is not necessarily limited to them: the exponential decay, the power-law decay, and no decay. Which function, each with the optimized parameter values, is most appropriate for given data is determined based on the log likelihood ratio statistic. We evaluate the parameter learning and the model selection methods through extensive experiments using synthetic data with two TDV models, one with the exponential decay and the other with power-low decay. We then applied the methods to the real opinion spreading data from a Japanese word-of-mouth communication site for cosmetics using aforementioned three decay functions, and show that most opinions conform to the TDV model of the power-law decay function.

The paper is organized as follows. We define the TDV model in Section 2 and explain how the model parameters are learned and the most appropriate model is selected in Section 3. The performance of parameter learning and model selection using the synthetic data is reported in Section 4 and the finding from the analysis of real data is reported in Section 5. We end this paper by summarizing the main result in Section 6.

2 Voter Model with Temporal Decay Dynamics

We define the TDV (Temporal Decay Voter) model. Let $G = (V, E)$ be a directed network with self-loops, where V and E ($\subset V \times V$) are the sets of all nodes and links in the network, respectively. Here, $(u, v) \in E$ denotes a (directed) link from node u to node v. When there is a link (u, v), we assume that v can be influenced by its neighbor u in opinion formation process. For a node $v \in V$, let $B(v)$ denote the set of neighbors of v in G, that is,

$$B(v) \ = \ \{u \in V;\ (u, v) \in E\}.$$

Note that $v \in B(v)$. Given an integer K with $K \geq 2$, we consider the spread of K opinions (opinion $1, \cdots,$ opinion K) on G, where each node holds exactly one of the K opinions at any time t (≥ 0). We assume that each node of G initially holds one of the K opinions with equal probability at time $t = 0$. We denote by

$$g_t : V \to \{1, \cdots, K\}$$

the *opinion distribution* at time t, where $g_t(v)$ stands for the opinion of node v at time t. Note that g_0 stands for the initial opinion distribution. For any $v \in V$ and $k \in \{1, 2, \cdots, K\}$, let $U_k(t, v)$ be the set of v's neighbors that hold opinion k as its latest opinion (before time t), i.e.,

$$U_k(t, v) \ = \ \{u \in B(v);\ \varphi_t(u) = k\},$$

where $\varphi_t(u)$ is the latest opinion of u (before time t).

2.1 Voter Model

We first recall the definition of the voter model (see, e.g., [13]), which is one of the standard models of opinion dynamics, where K is usually set to 2. The evolution process of the voter model is defined as follows:

1. At time 0, each node v independently decides its update time t according to some probability distribution such as an exponential distribution with parameter $\gamma_v = 1$.[1] The successive update time is determined similarly at each update time t.
2. At an update time t, the node v adopts the opinion of a randomly chosen neighbor u, i.e.,
$$g_t(v) \ = \ \varphi_t(u).$$
3. The process is repeated from the initial time $t = 0$ until the next update-time passes a given final-time T_1.

We note that in the voter model each individual tends to adopt the majority opinion among its neighbors.[2] Here note that the definition of one's neighbors include oneself because of the existence of self loop. Thus, we can extend the original voter model with 2 opinions to a voter model with K opinions by replacing Step 2 with: At an update time t, the node v selects one of the K opinions according to the probability distribution,

$$P(g_t(v) = k) \ = \ \frac{|U_k(t,v)|}{|B(v)|}, \quad (k = 1, \cdots, K). \tag{1}$$

2.2 Temporal Decay Voter Model

As mentioned earlier, people may decide their opinions by taking account not only of their neighbors' latest opinions, but also of their neighbors' past opinions including their own opinions. In order to model this kind of situation, for any $t > 0$ and $v \in V$, we consider the set $M(t, v)$ consisting of the time τ $(< t)$ at which an individual (a node) v manifested his/her opinion. For $k = 1, \cdots, K$, we also consider a subset of $M(t, v)$,

$$M_k(t,v) \ = \ \{\tau \in M(t,v);\ g_\tau(v) = k\},$$

where $M_k(t, v)$ is the set of node v's opinion manifestation time instances before time t in which v takes opinion k. Now, we can define a voter model which takes all the past opinions into consideration. In this model, Eq. (1) is replaced with

$$P(g_t(v) = k) \ = \ \frac{1 + \sum_{u \in B(v)} |M_k(t,u)|}{K + \sum_{u \in B(v)} |M(t,u)|}, \quad (k = 1, \cdots, K), \tag{2}$$

where we employed a Bayesian prior known as the Laplace smoothing. Here we note that the Laplace smoothing of Eq. (2) corresponds to the assumption that each node initially holds one of the K opinions with equal probability at time $t = 0$. Note also that the

[1] This assumes that the average delay time is 1.

[2] In reality there may be a case that one changes its opinion to a medium one (say 3) listening to two opposite opininons (say 1 and 5). The voter model does not consider this possibility unless at least one of the neighbors has already the medium opinion (3).

Laplace smoothing corresponds to a special case of Dirichlet distributions that are very often used as prior distributions in Bayesian statistics, and in fact the Dirichlet distribution is the conjugate prior of the categorical distribution and multinomial distribution. We refer to this voter model as the *base TDV model*.

Thus far, we assumed that all the past opinions are equally weighted. However, it is naturally conceivable that the quite old opinions have almost no influence. Older opinions are less influential in general. In order to reflect this kind of effects into the model, we consider introducing some decay functions. The simplest one is an exponential decay function defined by

$$\rho(\mathit{\Delta} t;\lambda) \;=\; \exp(-\lambda \mathit{\Delta} t), \tag{3}$$

where $\lambda \geq 0$ is a parameter and $\mathit{\Delta} t = t - \tau$ stands for the time difference between the opinion adoption time t and the opinion manifestation time τ. Another natural one would be a power-law decay function defined by

$$\rho(\mathit{\Delta} t;\lambda) \;=\; (\mathit{\Delta} t)^{-\lambda} \;=\; \exp(-\lambda \log \mathit{\Delta} t), \tag{4}$$

where $\lambda \geq 0$ is a parameter.

Now, we construct a more general decay function. For a given positive integer J, let $f_1(\mathit{\Delta} t), \cdots, f_J(\mathit{\Delta} t)$ be functions on $(0, +\infty)$ such that 1, $f_1(\mathit{\Delta} t), \cdots, f_J(\mathit{\Delta} t)$ are linearly independent, that is, if $\lambda_0, \lambda_1, \cdots, \lambda_J$ are real numbers and satisfy

$$\lambda_0 + \sum_{j=1}^{J} \lambda_j f_j(\mathit{\Delta} t) \;=\; 0, \quad (\forall \mathit{\Delta} t \in (0, +\infty)),$$

then $\lambda_0 = \lambda_1 = \cdots = \lambda_J = 0$. We then consider a J-dimensional feature vector,

$$\boldsymbol{F}_J(\mathit{\Delta} t) \;=\; (f_1(\mathit{\Delta} t), \cdots, f_J(\mathit{\Delta} t))^T,$$

where $\boldsymbol{a}^T$ denote the transpose of column vector $\boldsymbol{a}$. For a J-dimensional real column vector with non-negative elements,

$$\boldsymbol{\lambda}_J \;=\; (\lambda_1, \cdots, \lambda_J)^T,$$

which is a parameter vector, we define a decay function $\rho(\mathit{\Delta} t; \boldsymbol{\lambda}_J)$ by

$$\rho(\mathit{\Delta} t;\boldsymbol{\lambda}_J) \;=\; \exp\left(-\boldsymbol{\lambda}_J{}^T \boldsymbol{F}_J(\mathit{\Delta} t)\right), \tag{5}$$

where the matrix operations are used. Representative candidates of feature vector $\boldsymbol{F}_J(\mathit{\Delta} t)$ include

$$\boldsymbol{F}_1(\mathit{\Delta} t) = \mathit{\Delta} t, \quad \boldsymbol{F}_1(\mathit{\Delta} t) = \log \mathit{\Delta} t, \quad \boldsymbol{F}_1(\mathit{\Delta} t) = (\mathit{\Delta} t)^2$$

for $J = 1$,

$$\boldsymbol{F}_2(\mathit{\Delta} t) = (\mathit{\Delta} t,\, \log \mathit{\Delta} t)^T, \quad \boldsymbol{F}_2(\mathit{\Delta} t) = \left(\mathit{\Delta} t,\, (\mathit{\Delta} t)^2\right)^T, \quad \boldsymbol{F}_2(\mathit{\Delta} t) = \left(\log \mathit{\Delta} t,\, (\mathit{\Delta} t)^2\right)^T$$

for $J = 2$,

$$\boldsymbol{F}_3(\mathit{\Delta} t) \;=\; \left(\mathit{\Delta} t,\, \log \mathit{\Delta} t,\, (\mathit{\Delta} t)^2\right)^T$$

for $J = 3$, etc. Note that $\rho(\varDelta t; \boldsymbol{\lambda}_J)$ becomes the exponential decay function if $J = 1$ and $\boldsymbol{F}_J(\varDelta t) = \varDelta t$, and the power-law decay function if $J = 1$ and $\boldsymbol{F}_J(\varDelta t) = \log \varDelta t$.

Using our general decay function $\rho(\varDelta t; \boldsymbol{\lambda}_J)$ (see Eq. (5)), we define the TDV (Temporal Decay Voter) model in the following way. In this model, Eq. (1) is replaced with

$$P(g_t(v) = k) \; = \; \frac{1 + \sum_{u \in B(v)} \sum_{\tau \in M_k(t,u)} \rho(t - \tau; \boldsymbol{\lambda}_J)}{K + \sum_{u \in B(v)} \sum_{\tau \in M(t,u)} \rho(t - \tau; \boldsymbol{\lambda}_J)}, \quad (k = 1, \cdots, K). \tag{6}$$

Here note that Eq. (6) is reduced to Eq. (2) when $\boldsymbol{\lambda}_J$ is the J-dimensional zero-vector $\mathbf{0}_J$, that is, the TDV model of $\boldsymbol{\lambda}_J = \mathbf{0}_J$ coincides with the base TDV model.

3 Learning Method

We consider the problem of identifying the TDV model on network G from an observed data $\mathcal{D}_{T_0}$ in time-span $[0, T_0]$, where $\mathcal{D}_{T_0}$ consists of a sequence of (k, t, v) such that node v changed its opinion to opinion k at time t for $0 \leq t \leq T_0$. The identified model can be used to predict how much of the share each opinion will have at a future time T_1 ($> T_0$), and to identify both high decay tendency data sets and low decay tendency data sets.

3.1 Parameter Estimation

We describe a method for estimating decay parameter values of the TDV model from a given observed opinion spreading data $\mathcal{D}_{T_0}$. Based on the evolution process of our model (see Eq. (6)), we can obtain the likelihood function,

$$\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{\lambda}_J) \; = \; \log \left(\prod_{(k,t,v) \in \mathcal{D}_{T_0}} P(g_t(v) = k) \right), \tag{7}$$

where $\boldsymbol{\lambda}_J$ stands for the J-dimensional vector of decay parameter values, as explained in the previous subsection. Thus our estimation problem is formulated as a maximization problem of the objective function $\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{\lambda}_J)$ with respect to $\boldsymbol{\lambda}_J$.

We derive an iterative algorithm for obtaining the maximum likelihood estimators. From the definitions of $P(g_t(v) = k)$ (see Eq. (6)) and $\rho(\varDelta t; \boldsymbol{\lambda}_J)$ (see Eq. (5)), we can express Eq. (7) as follows:

$$\begin{aligned} \mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{\lambda}_J) = & \sum_{(k,t,v) \in \mathcal{D}_{T_0}} \log \left(1 + \sum_{u \in B(v)} \sum_{\tau \in M_k(t,u)} \exp\left(-\boldsymbol{\lambda}_J{}^T \boldsymbol{F}_J(t - \tau) \right) \right) \\ & - \sum_{(k,t,v) \in \mathcal{D}_{T_0}} \log \left(K + \sum_{u \in B(v)} \sum_{\tau \in M(t,u)} \exp\left(-\boldsymbol{\lambda}_J{}^T \boldsymbol{F}_J(t - \tau) \right) \right). \end{aligned} \tag{8}$$

Now, let $\bar{\boldsymbol{\lambda}}_J$ be the current estimate of $\boldsymbol{\lambda}_J$. We foucus on the first term of the right-hand side of Eq. (8), and define $q_{k,t,v}(\tau; \boldsymbol{\lambda}_J)$ by

$$q_{k,t,v}(\tau; \boldsymbol{\lambda}_J) \; = \; \frac{\exp\left(-\boldsymbol{\lambda}_J{}^T \boldsymbol{F}_J(t - \tau) \right)}{1 + \sum_{u \in B(v)} \sum_{\tau' \in M_k(t,u)} \exp\left(-\boldsymbol{\lambda}_J{}^T \boldsymbol{F}_J(t - \tau') \right)}$$

for any $k \in \{1, \cdots, K\}$, $t \in (0, T]$, $v \in V$, and $\tau \in \bigcup_{u \in B(v)} M_k(t, u)$. Note that for any $(k, t, v) \in \mathcal{D}_{T_0}$,

$$q_{k,t,v}(\tau; \boldsymbol{\lambda}_J) > 0, \quad \left(\forall \tau \in \bigcup_{u \in B(v)} M_k(t, u)\right), \tag{9}$$

$$\sum_{u \in B(v)} \sum_{\tau \in M_k(t,u)} q_{k,t,u}(\tau; \boldsymbol{\lambda}_J) + \frac{1}{1 + \sum_{u \in B(v)} \sum_{\tau' \in M_k(t,u)} \exp\left(-\boldsymbol{\lambda}_J{}^T \boldsymbol{F}_J(t - \tau')\right)} = 1. \tag{10}$$

We can transform our objective function as follows:

$$\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{\lambda}_J) \;=\; Q\left(\boldsymbol{\lambda}_J; \bar{\boldsymbol{\lambda}}_J\right) - \mathcal{H}\left(\boldsymbol{\lambda}_J; \bar{\boldsymbol{\lambda}}_J\right), \tag{11}$$

where $Q\left(\boldsymbol{\lambda}_J; \bar{\boldsymbol{\lambda}}_J\right)$ is defined by

$$\begin{aligned} Q\left(\boldsymbol{\lambda}_J; \bar{\boldsymbol{\lambda}}_J\right) = &- \sum_{(k,t,v) \in \mathcal{D}_{T_0}} \sum_{u \in B(v)} \sum_{\tau \in M_k(t,u)} q_{k,t,v}\left(\tau; \bar{\boldsymbol{\lambda}}_J\right) \boldsymbol{\lambda}_J{}^T \boldsymbol{F}_J(t - \tau) \\ &- \sum_{(k,t,v) \in \mathcal{D}_{T_0}} \log\left(K + \sum_{u \in B(v)} \sum_{\tau \in M(t,u)} \exp\left(-\boldsymbol{\lambda}_J{}^T \boldsymbol{F}_J(t - \tau)\right)\right), \end{aligned} \tag{12}$$

and $\mathcal{H}\left(\boldsymbol{\lambda}_J; \bar{\boldsymbol{\lambda}}_J\right)$ is defined by

$$\begin{aligned} \mathcal{H}\left(\boldsymbol{\lambda}_J; \bar{\boldsymbol{\lambda}}_J\right) \;=\; \sum_{(k,t,v) \in \mathcal{D}_{T_0}} & \left\{ \sum_{u \in B(v)} \sum_{\tau \in M_k(t,u)} q_{k,t,v}\left(\tau; \bar{\boldsymbol{\lambda}}_J\right) \log q_{k,t,v}\left(\tau; \boldsymbol{\lambda}_J\right) \right. \\ & + \frac{1}{1 + \sum_{u \in B(v)} \sum_{\tau' \in M_k(t,u)} \exp\left(-\bar{\boldsymbol{\lambda}}_J{}^T \boldsymbol{F}_J(t - \tau')\right)} \\ & \left. \times \log\left(\frac{1}{1 + \sum_{u \in B(v)} \sum_{\tau' \in M_k(t,u)} \exp\left(-\boldsymbol{\lambda}_J{}^T \boldsymbol{F}_J(t - \tau')\right)}\right) \right\}. \end{aligned} \tag{13}$$

By Eqs. (9), (10), (13), and the property of the KL-divergence, it turns out that $\mathcal{H}\left(\boldsymbol{\lambda}_J; \bar{\boldsymbol{\lambda}}_J\right)$ is maximized at $\boldsymbol{\lambda}_J = \bar{\boldsymbol{\lambda}}_J$. Hence, we can increase the value of $\mathcal{L}(\mathcal{D}_{T_0}; \boldsymbol{\lambda}_J)$ by maximizing $Q\left(\boldsymbol{\lambda}_J; \bar{\boldsymbol{\lambda}}_J\right)$ with respect to $\boldsymbol{\lambda}_J$ (see Eq. (11)).

We derive an update formula for maximizing $Q(\boldsymbol{\lambda}_J; \bar{\boldsymbol{\lambda}}_J)$. We foucus on the second term of the right-hand side of Eq. (12) (see also the second term of the right-hand side of Eq. (8)), and define $r_{t,v}(\tau; \boldsymbol{\lambda}_J)$ by

$$r_{t,v}(\tau; \boldsymbol{\lambda}_J) \;=\; \frac{\exp\left(-\boldsymbol{\lambda}_J{}^T \boldsymbol{F}_J(t - \tau)\right)}{K + \sum_{u \in B(v)} \sum_{\tau' \in M(t,u)} \exp\left(-\boldsymbol{\lambda}_J{}^T \boldsymbol{F}_J(t - \tau')\right)} \tag{14}$$

for any $t \in (0, T]$, $v \in V$, and $\tau \in \bigcup_{u \in B(v)} M(t, u)$. Note that for any $(k, t, v) \in \mathcal{D}_{T_0}$,

$$r_{t,v}(\tau; \boldsymbol{\lambda}_J) > 0, \quad \left(\forall \tau \in \bigcup_{u \in B(v)} M(t, u)\right), \qquad \sum_{u \in B(v)} \sum_{\tau \in M(t,u)} r_{t,u}(\tau; \boldsymbol{\lambda}_J) < 1. \tag{15}$$

From Eqs. (12) and (14), we can easily see that the gradient vector of $Q\left(\boldsymbol{\lambda}_J;\bar{\boldsymbol{\lambda}}_J\right)$ with respect to $\boldsymbol{\lambda}_J$ is given by

$$\frac{\partial Q\left(\boldsymbol{\lambda}_J;\bar{\boldsymbol{\lambda}}_J\right)}{\partial \boldsymbol{\lambda}_J} = -\sum_{(t,v,k)\in\mathcal{D}_{T_0}} \sum_{u\in B(v)} \left(\sum_{\tau\in M_k(t,u)} q_{t,v,k}\left(\tau;\bar{\boldsymbol{\lambda}}_J\right) \boldsymbol{F}_J(t-\tau) - \sum_{\tau\in M(t,u)} r_{t,v}(\tau;\boldsymbol{\lambda}_J)\boldsymbol{F}_J(t-\tau)\right). \qquad (16)$$

Moreover, from Eqs. (14) and (16), we can obtain the Hessian matrix of $Q\left(\boldsymbol{\lambda}_J;\bar{\boldsymbol{\lambda}}_J\right)$ as follows:

$$\frac{\partial^2 Q\left(\boldsymbol{\lambda}_J;\bar{\boldsymbol{\lambda}}_J\right)}{\partial \boldsymbol{\lambda}_J \partial \boldsymbol{\lambda}_J{}^T} = -\sum_{(k,t,v)\in\mathcal{D}_{T_0}} \left\{ \sum_{u\in B(v)} \sum_{\tau\in M(t,u)} r_{t,v}(\tau;\boldsymbol{\lambda}_J)\, \boldsymbol{F}_J(t-\tau)\, \boldsymbol{F}_J(t-\tau)^T - \left(\sum_{u\in B(v)} \sum_{\tau\in M(t,u)} r_{t,v}(\tau;\boldsymbol{\lambda}_J)\, \boldsymbol{F}_J(t-\tau) \right) \left(\sum_{u\in B(v)} \sum_{\tau\in M(t,u)} r_{t,v}(\tau;\boldsymbol{\lambda}_J)\, \boldsymbol{F}_J(t-\tau) \right)^T \right\}. \qquad (17)$$

By Eq. (17), for any J-dimensional real column vector $\boldsymbol{x}_J$, we have

$$\begin{aligned}
&\boldsymbol{x}_J{}^T \frac{\partial^2 Q\left(\boldsymbol{\lambda}_J;\bar{\boldsymbol{\lambda}}_J\right)}{\partial \boldsymbol{\lambda}_J \partial \boldsymbol{\lambda}_J{}^T} \boldsymbol{x}_J \\
&= -\sum_{(k,t,v)\in\mathcal{D}_{T_0}} \left\{ \sum_{u\in B(v)} \sum_{\tau\in M(t,u)} r_{t,v}(\tau;\boldsymbol{\lambda}_J) \left(\boldsymbol{x}_J{}^T\, \boldsymbol{F}_J(t-\tau) \right)^2 - \left(\sum_{u\in B(v)} \sum_{\tau\in M(t,u)} r_{t,v}(\tau;\boldsymbol{\lambda}_J)\, \boldsymbol{x}_J{}^T\, \boldsymbol{F}_J(t-\tau) \right)^2 \right\} \\
&= -\sum_{(k,t,v)\in\mathcal{D}_{T_0}} \left\{ \sum_{u\in B(v)} \sum_{\tau\in M(t,u)} r_{t,v}(\tau;\boldsymbol{\lambda}_J) \left(\boldsymbol{x}_J{}^T\, \boldsymbol{F}_J(t-\tau) - \sum_{u\in B(v)} \sum_{\tau'\in M(t,u)} r_{t,v}(\tau';\boldsymbol{\lambda}_J)\, \boldsymbol{x}_J{}^T\, \boldsymbol{F}_J(t-\tau') \right)^2 \right. \\
&\qquad \left. + \left(1 - \sum_{u\in B(v)} \sum_{\tau\in M(t,u)} r_{t,v}(\tau;\boldsymbol{\lambda}_J) \right) \left(\sum_{u\in B(v)} \sum_{\tau\in M(t,u)} r_{t,v}(\tau;\boldsymbol{\lambda}_J)\, \boldsymbol{x}_J{}^T\, \boldsymbol{F}_J(t-\tau) \right)^2 \right\}.
\end{aligned}$$

Thus, by Eq. (15), we obtain

$$\boldsymbol{x}_J{}^T \frac{\partial^2 Q\left(\boldsymbol{\lambda}_J;\bar{\boldsymbol{\lambda}}_J\right)}{\partial \boldsymbol{\lambda}_J \partial \boldsymbol{\lambda}_J{}^T} \boldsymbol{x}_J \;\leq\; 0, \qquad \left(\forall \boldsymbol{x}_J \in \mathbf{R}^J\right),$$

that is, the Hessian matrix is negative semi-definite. Hence, by solving the equation

$$\frac{\partial Q\left(\boldsymbol{\lambda}_J; \bar{\boldsymbol{\lambda}}_J\right)}{\partial \boldsymbol{\lambda}_J} = \mathbf{0}_J$$

(see Eq. (16)), we can find the value of $\boldsymbol{\lambda}_J$ that maximizes $Q\left(\boldsymbol{\lambda}_J; \bar{\boldsymbol{\lambda}}_J\right)$. We employed a standard Newton Method in our experiments.

3.2 Model Selection

One of the important purposes of introducing the TDV model is to analyze how people are affected by their neighbors' past opinions for a specific opinion formation process. In what follows, for a given set of candidate decay functions (i.e., feature vectors), we consider selecting one being the most appropriate to the observed data $\mathcal{D}_{T_0}$ of $|\mathcal{D}_{T_0}| = N$, where N represents the number of opinion manifestations by individuals.

As mentioned in Section 2, the base TDV model is a special TDV model equipped with the decay function that equally weights all the past opinions. Thus, we first examine whether or not the TDV model equipped with a candidate decay function can be more appropriate to the observed data $\mathcal{D}_{T_0}$ than the base TDV model.[3] To this end, we employ the likelihood ratio test. For a given feature vector $\boldsymbol{F}_J(\Delta t)$, let $\hat{\boldsymbol{\lambda}}_J(\boldsymbol{F}_J)$ be the maximal likelihood estimator of the TDV model equipped with the decay function of $\boldsymbol{F}_J(\Delta t)$. Since the base TDV model is the TDV model of $\boldsymbol{\lambda}_J = \mathbf{0}_J$, the log-likelihood ratio statistic of the TDV model with $\boldsymbol{F}_J(\Delta t)$ against the base TDV model is given by

$$Y_N(\boldsymbol{F}_J) = \mathcal{L}\left(\mathcal{D}_{T_0}; \hat{\boldsymbol{\lambda}}_J(\boldsymbol{F}_J)\right) - \mathcal{L}\left(\mathcal{D}_{T_0}; \mathbf{0}_J\right). \tag{18}$$

It is well known that $2Y_N(\boldsymbol{F}_J)$ asymptotically approaches to the χ^2 distribution with J degrees of freedom as N increases. We set a significance level α $(0 < \alpha < 1)$, say $\alpha = 0.005$, and evaluate whether or not the TDV model with $\boldsymbol{F}_J(\Delta t)$ fits significantly better than the base TDV model by comparing $2Y_N(\boldsymbol{F}_J)$ to $\chi_{J,\alpha}$. Here, $\chi_{J,\alpha}$ denotes the upper α point of the χ^2 distribution of J degrees of freedom, that is, it is the positive number z such that

$$\frac{1}{\Gamma(J/2)2^{J/2}} \int_0^z y^{J/2-1} \exp\left(-\frac{y}{2}\right) dy = 1 - \alpha,$$

where $\Gamma(s)$ is the gamma function. We consider the set $\mathcal{FV}$ of the candidate feature vectors (i.e., decay functions) selected by this likelihood ratio test at significance level α. Next, we find the feature vector $\boldsymbol{F}^*_{J^*}(\Delta t) \in \mathcal{FV}$ such that it maximizes the log-likelihood ratio statistic $Y_N(\boldsymbol{F}_J)$, $(\boldsymbol{F}_J(\Delta t) \in \mathcal{FV})$, (see Eq. (18)), and propose selecting the TDV model equipped with the decay function of $\boldsymbol{F}^*_{J^*}(\Delta t)$. If the set $\mathcal{FV}$ is empty, we select the base TDV model for $\mathcal{D}_{T_0}$.

[3] The base TDV model is not the only baseline model with which the proposed method is to be compared. The simplest one would be the random opininon model in which each user chooses its opinionn randomly independent of its neighbors.

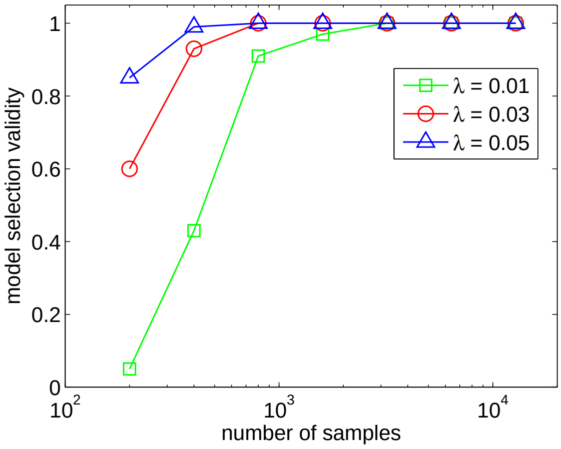


Fig. 1. Results of model selection validity for the exponential TDV model.

Here we recall that typical decay functions in natural and social sciences include the exponential decay function (see Eq. (3)) and the power-law decay functions (see Eq. (4)). We refer to the TDV models of the exponential and the power-law decay functions as the *exponential TDV model* and the *power-law TDV model*, respectively. In our experiments, we in particular focus on investigating which of the base, the exponential, and the power-law TDV models best fits to the observed data $\mathcal{D}_{T_0}$. Thus, the TDV model to be considered has $J = 1$ and parameter λ.

4 Evaluation by Synthetic Data

Using synthetic data, we examined the effectiveness of the proposed method for parameter estimation and model selection. We assumed complete networks for simplicity. According to the TDV model, we artificially generated an opinion diffusion sequence $\mathcal{D}_{T_0}$ consisting of 3-tuple (k, t, v) of opinion k, time t and node v such that $|\mathcal{D}_{T_0}| = N$, and applied the proposed method to the observed data $\mathcal{D}_{T_0}$, where the significance level $\alpha = 0.005$ was used for model selection. As mentioned in the previous section, we assumed two cases where the true decay follows the exponential distribution (see Eq. (3)) and the power-law distribution (see Eq. (3)), respectively. Let Y_N^e and Y_N^p denote the log-likelihood ratio statistics of the exponential and the power-law TDV models against the base TDV model, respectively (see Eq. (18)). We varied the value of parameter λ in the following range: $\lambda = 0.01, 0.03, 0.05$ for the exponential TDV model, and $\lambda = 0.4, 0.5, 0.6$ for the power-law TDV model, on the basis of the analysis performed for the real world @cosme dataset (see, Section 5). We conducted 100 trials varying the observed data $\mathcal{D}_{T_0}$ of $|\mathcal{D}_{T_0}| = N$, and evaluated the proposed method.

First, we investigated the model selection validity $\mathcal{F}_N/100$, where $\mathcal{F}_N$ is the number of trials in which the true model was successfully selected by the proposed method.

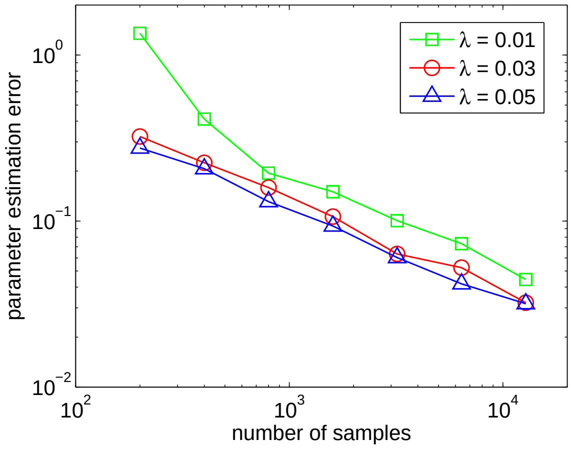


Fig. 2. Results of Parameter estimation error for the exponential TDV model.

Namely, if the exponential TDV model is the true model, then $\mathcal{F}_N$ is defined by the number of trials such that

$$2Y_N^e \ > \ \max\left(\chi_{1,\alpha},\, 2Y_N^p\right),$$

and if the power-law TDV model is the true model, then $\mathcal{F}_N$ is defined by the number of trials such that

$$2Y_N^p \ > \ \max\left(\chi_{1,\alpha},\, 2Y_N^e\right).$$

Second, we examined the parameter estimation error $\mathcal{E}_N$ for the trials in which the true model was selected by the proposed method. Here, $\mathcal{E}_N$ is defined by

$$\mathcal{E}_N = \frac{|\hat{\lambda}(N) - \lambda^*|}{\lambda^*},$$

where λ^* is the true value of parameter λ, and $\hat{\lambda}(N)$ is the value estimated by the proposed method from the observed data $\mathcal{D}_{T_0}$ of $|\mathcal{D}_{T_0}| = N$. Figures 1 and 2 show the results for the exponential TDV model, and Figures 3 and 4 show the results for the power-law TDV model. Here, Figures 1 and 3 display model selection validity $\mathcal{F}_N/100$ as a function of sample size N. Figures 2 and 4 display parameter estimation error $\mathcal{E}_N$ as a function of sample size N. As expected, $\mathcal{F}_N$ increases and $\mathcal{E}_N$ decreases as N increases. Moreover, as λ becomes larger, $\mathcal{F}_N$ increases and $\mathcal{E}_N$ decreases. Note that a large λ means quickly forgetting past activities, and a small λ means slowly forgetting them. Thus, we can consider that a TDV model of smaller λ requires more samples to correctly learn the model. From Figures 1, 2, 3 and 4, we observe that the proposed method can work almost perfectly when N is greater than 500, and λ is greater than 0.01 for the exponential TDV model and greater than 0.4 for the power-law TDV model.

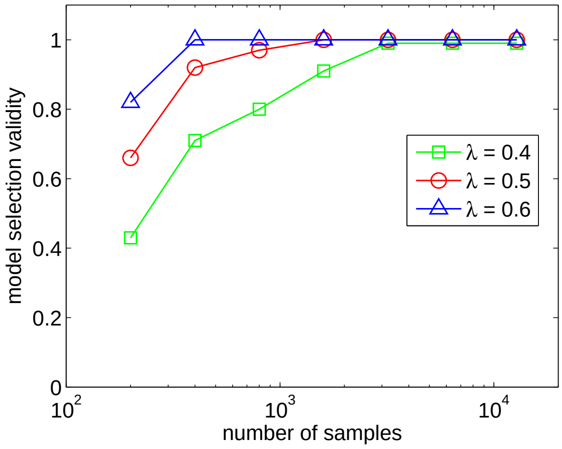


Fig. 3. Results of model selection validity for the power-law TDV model.

5 Findings in Opinion Formation on Social Media

5.1 Dataset

We collected real data from "@cosme" [4], which is a Japanese word-of-mouth communication website for cosmetics. In @cosme, a user can post a review and give a score of each brand (one from 1 to 7). When one user registers another user as his/her favorite user, a "fan-link" is created between them. We traced up to ten steps in the fan-links from a randomly chosen user in December 2009, and collected a set of (b, k, t, v)'s, where (b, k, t, v) means that user v scored brand b k points at time t. The number of brands was 7,139, the number of users was 45,024, and the number of reviews posted was 331,084. For each brand b, we regarded the point k scored by a user v as the opinion k of v, and constructed the opinion diffusion sequence $\mathcal{D}_{T_0}(b)$ consisting of 3-tuple (k, t, v). In particular, we focused on these brands in which the number of samples $N = |\mathcal{D}_{T_0}(b)|$ was greater than 500. Then, the number of brands was 120. We refer to this dataset as the @cosme dataset.

5.2 Results

We applied the proposed method to the @cosme dataset. Again, we adopted the temporal decay voter models with the exponential and the power-law distributions, and used the significance level $\alpha = 0.005$ for model selection. There were 9 brands such that $2Y_N^e > \chi_{1,\alpha}$, and 93 brands such that $2Y_N^p > \chi_{1,\alpha}$. Here, in the same way as the previous section, Y_N^e and Y_N^p denote the log-likelihood ratio statistics of the exponential and the power-law TDV models against the base TDV model, respectively. Further, there were

[4] `http://www.cosme.net/`

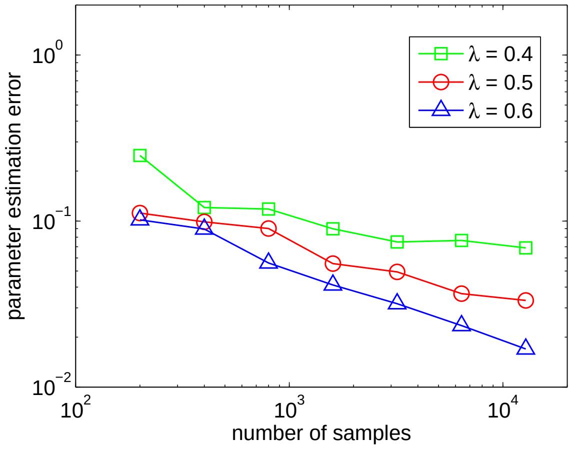


Fig. 4. Results of Parameter estimation error for the power-law TDV model.

92 brands such that $2Y_N^p > \max\left(\chi_{1,\alpha}, 2Y_N^e\right)$, one brand such that $2Y_N^e > \max\left(\chi_{1,\alpha}, 2Y_N^p\right)$, and 27 brands such that $\max\left(2Y_N^p, 2Y_N^e\right) \leq \chi_{1,\alpha}$. Namely, according to the proposed method, 92 brands were the power-law TDV model, 27 brands were the base TDV model, and only one brand was the exponential TDV model. These results show that most brands conform to the power-law TDV model. This also agrees with the work [1, 19] that many human actions are related to power-laws.

Figures 5 and 6 show the results for the @cosme dataset from the point of view of the power-law TDV model. Figure 5 plots the log-likelihood ratio statistic Y_N^p for each brand as a function of sample size N, where the thick solid line indicates the value of $\chi_{i,\alpha}$. In addition to the brands plotted, there is a brand such that $Y_N^p = Y_N^e = 0$. It was brand “YOJIYA”, which is a traditional Kyoto brand, and is known as a brand releasing new products less frequently. Thus, we speculate that it conforms to the base TDV model. Figure 6 plots the pair $\left(Y_N^p, \hat{\lambda}(N)\right)$ for the brands in which the power-law TDV model was selected by the proposed method, where $\hat{\lambda}(N)$ is the value of parameter λ estimated by the proposed method from the observed data $\mathcal{D}_{T_0}(b)$ of $|\mathcal{D}_{T_0}(b)| = N$. From Figure 6, we observe that Y_N^p and $\hat{\lambda}(N)$ are positively correlated. This agrees with the fact that the power-law TDV model with $\lambda = 0$ corresponds to the base TDV model. In Figures 5 and 6, the big solid red circle indicates the brand “LUSH-JAPAN”, which had the largest values of Y_N^p, $\hat{\lambda}(N)$ and N, respectively. We also find the big solid green triangle in Figure 5 as a brand that had a large value of Y_N^p and a relatively small value of N. This was the brand “SHISEIDO ELIXIR SUPERIEUR”, which had the seventh largest value of Y_N^p, $N = 584$, and $\hat{\lambda}(N) = 0.58$. Note that these brands “LUSH-JAPAN” and “SHISEIDO ELIXIR SUPERIEUR” are known as brands that were recently established and release new products frequently. Thus, we speculate that they conform to the power-law TDV model with large λ.

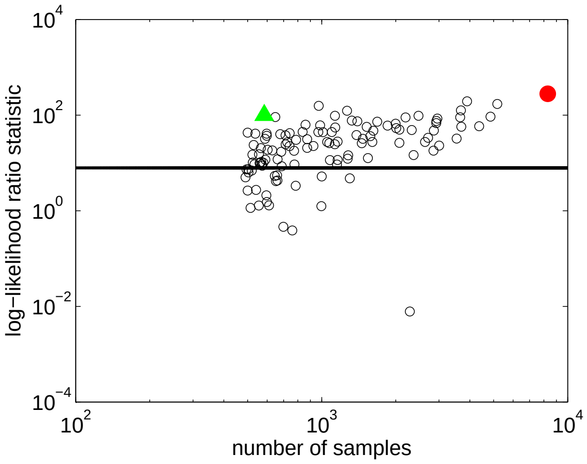


Fig. 5. Log-likelihood ratio statistic Y_N^p and number of samples N for the @cosme dataset.

6 Conclusion

We addressed the problem of how people make their own decisions based on their neighbors' opinions. The model best suited to discuss this problem is the voter model and several variants of this model have been proposed and used extensively. However, all of these models assume that people use their neighbors' latest opinions. People change opinions over time and some opinions are more persistent and some others are less persistent. These depend on many factors but the existing models do not take this effect into consideration. In this paper, we, in particular, addressed the problem of how people's opinions are affected by their own and other peoples' opinion histories. It would be reasonable to assume that older opinions are less influential and recent ones are more influential. Based on this assumption, we devised a new voter model, called the temporal decay voter (TDV) model which uses all the past opinions in decision making in which decay is assumed to be a linear combination of representative decay functions each with different decay factors. The representative functions include the linear decay, the exponential decay, the power-law decay and many more. Each of them specifies only the form and the parameters remain unspecified. We formulated this as a machine learning problem and solved the following two problems: 1) Given the observed sequence of people's opinion manifestation and an assumed decay function, learn the parameter values of the function such that the corresponding TDV model best explains the observation, and 2) Given a set of decay functions each with the optimal parameter values, choose the best model and refute others. We solved the former problem by maximizing the likelihood and derived an efficient parameter updating algorithm, and the latter problem by choosing the decay model that maximizes the log likelihood ratio statistic. We first tested the proposed algorithms by synthetic datasets assuming that there are two decay models: the exponential decay and the power-law decay. We confirmed that the learning algorithm correctly identifies the parameter values and the model selection

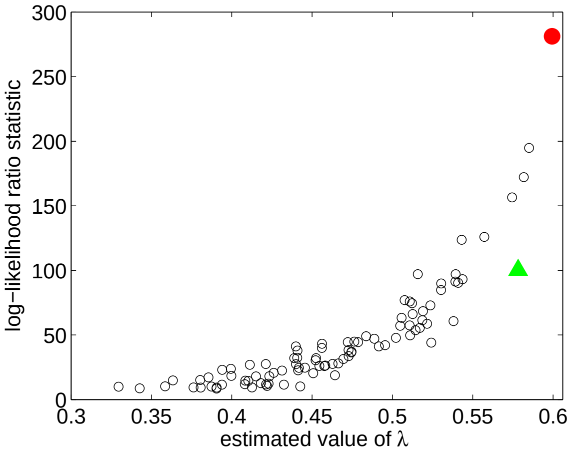


Fig. 6. Log-likelihood ratio statistic Y_N^p and estimated parameter value $\hat{\lambda}(N)$ for the @cosme dataset.

algorithm correctly identifies which model the data came from. We then applied the method to the real opinion diffusion data taken from a Japanese word-of-mouth communication site for cosmetics. We used the two decay functions above and added no decay function as a baseline. The result of the analysis revealed that opinions of most of the brands conform to the TDV model of the power-law decay function. We found this interesting because this is consistent with the observation that many human actions are related to the power-law. Some brands showed behaviors characteristic to the brands, e.g., the older brand that releases new product less frequently naturally follows no decay TDV and the newer brand that releases new product more frequently naturally follows the power-law decay TDV with large decay constant, which are all well interpretable.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-11-4111, and JSPS Grant-in-Aid for Scientific Research (C) (No. 23500194).

References

1. Barabási, A.L.: The origin of bursts and heavy tails in human dynamics. Nature 435, 207–211 (2005)
2. Castellano, C., Munoz, M.A., Pastor-Satorras, R.: Nonlinear *q*-voter model. Physical Review E 80, Article 041129 (2009)

3. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09). pp. 199–208 (2009)
4. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of the 10th IEEE International Conference on Data Mining (ICDM'10). pp. 88–97 (2010)
5. Crandall, D., Cosley, D., Huttenlocner, D., Kleinberg, J., Suri, S.: Feedback effects between similarity and social influence in online communities. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08). pp. 160–168 (2008)
6. Domingos, P.: Mining social networks for viral marketing. IEEE Intelligent Systems 20, 80–82 (2005)
7. Even-Dar, E., Shapira, A.: A note on maximizing the spread of influence in social networks. In: Proceedings of the 3rd International Workshop on Internet and Network Economics (WINE'07). pp. 281–286 (2007)
8. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations 6, 43–52 (2004)
9. Holme, P., Newman, M.E.J.: Nonequilibrium phase transition in the coevolution of networks and opinions. Physical Review E 74, Article 056108 (2006)
10. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03). pp. 137–146 (2003)
11. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Mining and Knowledge Discovery 20, 70–97 (2010)
12. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Learning to predict opinion share in social networks. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI'10). pp. 1364–1370 (2010)
13. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Detecting anti-majority opinionists using value-weighted mixture voter model. In: Proceedings of the 14th International Conference on Discovery Science (DS'11). pp. 150–164 (2011)
14. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09). pp. 447–456 (2009)
15. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. ACM Transactions on the Web 1, Article 5 (2007)
16. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07). pp. 420–429 (2007)
17. Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45, 167–256 (2003)
18. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E 66, Article 035101 (2002)
19. Oliveira, J.G., Barabási, A.L.: Dawin and Einstein correspondence patterns. Nature 437, 1251 (2005)
20. Sood, V., Redner, S.: Voter model on heterogeneous graphs. Physical Review Letters 94, Article 178701 (2005)
21. Wu, F., Huberman, B.A.: How public opinion forms. In: Proceedings of the 4th International Workshop on Internet and Network Economics (WINE'08). pp. 334–341 (2008)
22. Yang, H., Wu, Z., Zhou, C., Zhou, T., Wang, B.: Effects of social diversity on the emergence of global consensus in opinion dynamics. Physical Review E 80, Article 046108 (2009)

Burst Detection in a Sequence of Tweets based on Information Diffusion Model

Kazumi Saito[1], Kouzou Ohara[2], Masahiro Kimura[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp

[2] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 252-5258, Japan
ohara@it.aoyama.ac.jp

[3] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp

[4] Institute of Scientific and Industrial Research, Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We propose a method of detecting the period in which a burst of information diffusion took place from an observed diffusion sequence data over a social network and report the results obtained by applying it to the real Twitter data. We assume a generic information diffusion model in which time delay associated with the diffusion follows the exponential distribution and the burst is directly reflected to the changes in the time delay parameter of the distribution (inverse of the average time delay). The shape of the parameter change is approximated by a series of step functions and the problem of detecting the change points and finding the values of the parameter is formulated as an optimization problem of maximizing the likelihood of generating the observed diffusion sequence. Time complexity of the search is almost proportional to to the number of observed data points (possible change points) and very efficient. We apply the method to the real Twitter data of the 2011 To-hoku earthquake and tsunami, and show that the proposed method is by far efficient than a naive method that adopts exhaustive search, and more accurate than a simple greedy method. Two interesting discoveries are that a burst period between two change points detected by the proposed method tends to contain massive homogeneous tweets on a specific topic even if the observed diffusion sequence consists of heterogeneous tweets on various topics, and that assuming the information diffusion path is a line shape tree can give a good approximation of the maximum likelihood estimator when the actual diffusion path is not known.

1 Introduction

Recent technological innovation and popularization of high performance mobile/smart phones has changed our communication style drastically and the use of various social media such as Twitter and Facebook has been affecting our daily lives substantially. In these social media, information propagates through the social network formed based on friendship relations. Especially, Twitter, micro-blog in which the number of characters is limited to 140, is now very popular among the young generation due to its handiness and easiness of usage, and it is fresh to our memory that Twitter played a very important role as the information infrastructure during the recent natural disaster, both domestic and abroad, including the 2011 To-hoku earthquake and tsunami in Japan.

In these social networks, there have been proposed several measures, called centrality, that characterize nodes in the network based on the structure of the network [11, 1, 3]. While such centrality measures can be used to identify those nodes that play an important role in diffusing information over the network, it has also been shown that measures based solely on the network structure are not good enough to a such problem of influence maximization [11, 1, 3] in which the task is to identify a limited number of nodes which together maximizes the information spread and that explicit use of information diffusion mechanism is essential [5]. In general, the mechanism is represented by a probabilistic diffusion model. Most representative and basic ones are the Independent Cascade (IC) model [2, 4] and the Linear Threshold (LT) model [12, 13] including their extended versions that explicitly handle asynchronous time delay, Asynchronous time delay Independent Cascade (AsIC) model [8] and Asynchronous time delay Linear Threshold (AsLT) model [9]. In fact, the nodes and links that are identified to be influential using these models are substantially different from those identified by the existing centrality measures.

In reality, we observe that the information on a certain topic propagates explosively for a very short period of time. Because such information affects our behaviour strongly, it is important to understand the observed event in a timely manner. This brings in an important and interesting problem which is to accurately and efficiently detect the burst from the observed information diffusion data and to identify what caused this burst and how long it persisted. Any of the above mentioned probabilistic models cannot handle this kind of problem because they assume that information diffuses in a stationary environment, i.e. model parameters are stationary. Zhu and Shasha [14] approached this problem without relying on a diffusion model. They detected a burst period for a target event by counting the number of its occurrences in a given time window and checking whether it exceeds a predetermined threshold or not. Kleinberg [6] challenged this problem using a hidden Markov model in which bursts appear naturally as state transitions, and successfully identified the hierarchical structure of e-mail messages. Sun et al. [10] extended Kleinberg's method so as to detect correlated burst patterns from multiple data streams that co-evolve over time.

We handle this problem by assuming that parameters in the diffusion model changed due to unknown external environmental factors and devise an efficient algorithm that accurately detects the changes in the parameter values from a single observed diffusion data sequence. In particular we note that the parameter related to the time delay is most crucial in the burst detection and focus on detecting the changes in the time delay parameter that defines the delay distribution. We modeled the time delay in AsIC and AsLT models by the exponential distribution, thus we do the same in this paper. This corresponds to associating the burst with the information diffusion with a shorter time delay. By focusing only on this time delay, we can devise a generic algorithm that does not depend on a specific information diffusion model, e.g. be it either AsIC or AsLT.

More precisely, we assume that time delay parameter changes are approximated by a series of step functions and propose an optimization algorithm that maximizes the likelihood ratio that is the ratio of the likelihood of observing the data assuming the time delay parameter changes (change points and parameter values between the successive change points) to the likelihood of observing the data assuming that there is no changes in the time delay parameter. The algorithm is based on iterative search based on recursive splitting with delayed backtracking, and requires no predetermined threshold. The time complexity is almost proportional to the number of observed data points (candidates of possible change points). We apply the method to the Twitter data observed during the 2011 To-hoku earthquake and tsunami and confirm that the proposed method can efficiently and accurately detect the change points. We further analyze the content of the tweets and report the discovery that even use of the diffusion sequence data of the same user ID (not necessarily the data on a specific topic) allows us to identify that a specific topic is talked intensively around the beginning of the period where the burst is detected, and the assumption we made that the information diffusion path is a line shape tree gives a good approximation of the maximum likelihood estimator in this problem setting. Finally, we discuss that although the detected change points do not correspond exactly to nodes in a social network that caused the burst period, the detected change points are useful to find such nodes because we can limit nodes to be considered by focusing on those around them.

The paper is organized as follows. Section 2 briefly describes the framework of information diffusion model on which our problem setting is based. Section 3 elucidates the problem setting, and Section 4 describes the change point detection method including two other methods that are used for comparison. Section 5 reports experimental results using real Twitter data. Section 6 summarizes what has been achieved in this work and addresses the future work.

2 Information Diffusion Model Framework

We consider information diffusion over a social network whose structure is defined as a directed graph $G = (V, E)$, where V and E $(\subset V \times V)$ represent a set of all nodes and a set of all links, respectively. Suppose that we observe a sequence of information diffusion $C = \{(v_0, t_0), (v_1, t_1), \cdots, (v_N, t_N)\}$ that arose from the information released at the source node v_0 at time t_0. Here, v_n is a node where the information has been propagated and t_n is its time. We assume that the time points are ordered such that $t_{n-1} < t_n$ for any $n \in \{1, \cdots N\}$. We further assume, as a standard setting, that the actual information diffusion paths of a sequence C correspond to a tree that is embedded in the directed graph G representing the social network[7], i.e., the parent node which passed the information to a node v_n is uniquely identified to be $v_{p(n)}$. Here, $p(n)$ is a function that returns the node identification number of the parent of the node v_n in the range of $\{0, \cdots, n-1\}$.

The information diffusion model we consider here is any model that explicitly incorporates the concept of asynchronous time delay such as AsIC model [8] and AsLT model [9] in contrast to the traditional IC model [2,4] and LT model [12,13] that do not consider the time delay. Said differently, it is a model that allows any real value for the time t_n at which the information has been propagated to a node v_n and assumes a certain probability distribution for the time delay $t_n - t_{p(n)}$. In this paper, we use the exponential distribution for the time delay, but any other distribution such as power law is feasible exactly in the same way.

3 Problem Settings

In this section we formally define the change point detection problem. As mentioned in Section 1, we assume that some unknown change took place in the course of information diffusion and what we observe is a sequence of information diffusion of some topic in which the change is encapsulated. Thus, our goal is to detect each change point and how long the change persisted from there. Note that we basically pay attention to a diffusion sequence of a certain topic. From our previous result that people's behaviors are quite similar when talking the same topic [8,9], we can assume that the time delay parameter $r_{u,v}$ which is in principle defined for each link $(u,v) \in E$ takes a uniform value regardless of the link it passes through. In other word, we set $r_{u,v} = r$ $(\forall (u,v) \in E)$ and thus, the time delay of information diffusion is represented by the following simple exponential distribution $p(t_n - t_{p(n)}; r) = r \exp(-r(t_n - t_{p(n)}))$.

With this preparation, we mathematically define the change point detection problem. Let's assume that we observe a set of time points of information diffusion sequence $\mathcal{D} = \{t_0, t_1, \cdots, t_N\}$. Let the time of the j-th change point be T_j $(t_0 < T_j < t_N)$. The delay parameter that the distribution follows switches from r_j to r_{j+1} at the j-th change point T_j. Namely, we are assuming a series of step functions as a shape of parameter changes. Let the set comprising J change points be $\mathcal{S}_J = \{T_1, \cdots, T_J\}$, and we set $T_0 = t_0$ and $T_{J+1} = t_N$ for the sake of convenience $(T_{j-1} < T_j)$. Let the division of $\mathcal{D}$ by $\mathcal{S}_J$ by $\mathcal{D}_j = \{t_n; T_{j-1} < t_n \le T_j\}$, i.e., $\mathcal{D} = \{t_0\} \cup \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_{J+1}$, and $|\mathcal{D}_j|$ represents the number of observed points in $(T_{j-1}, T_j]$. Here, we request that $|\mathcal{D}_j| \neq 0$ for any $j \in \{1, \cdots, J+1\}$ and there exists at least one t_n and $t_n \in \mathcal{D}_j$ is satisfied.

The log-likelihood for the $\mathcal{D}$, given a set of change points $\mathcal{S}_J$, is calculated, by defining the parameter vector $\mathbf{r}_{J+1} = (r_1, \cdots, r_{J+1})$, as follows.

$$\begin{aligned} L(\mathcal{D}; \mathbf{r}_{J+1}, \mathcal{S}_J) &= \log \prod_{j=1}^{J+1} \prod_{t_n \in \mathcal{D}_j} r_j \exp(-r_j(t_n - t_{p(n)})) \\ &= \sum_{j=1}^{J+1} |\mathcal{D}_j| \log r_j - \sum_{j=1}^{J+1} r_j \sum_{t_n \in \mathcal{D}_j} (t_n - t_{p(n)}). \end{aligned} \tag{1}$$

Thus, the maximum likelihood estimate of the parameter of Equation (1) is given by

$$\hat{r}_j^{-1} = \frac{1}{|\mathcal{D}_j|} \sum_{t_n \in \mathcal{D}_j} (t_n - t_{p(n)}), \quad j = 1, \cdots, J+1. \tag{2}$$

Further, substituting Equation (2) to Equation (1) leads to

$$L(\mathcal{D}; \hat{\mathbf{r}}_{J+1}, \mathcal{S}_J) = -N - \sum_{j=1}^{J+1} |\mathcal{D}_j| \log \left(\frac{1}{|\mathcal{D}_j|} \sum_{t_n \in \mathcal{D}_j} (t_n - t_{p(n)}) \right). \quad (3)$$

Therefore, the change point detection problem is reduced to the problem of finding the change point set $\mathcal{S}_J$ that maximizes Equation (3). However, Equation (3) alone does not allow us to directly evaluate the effect of introducing $\mathcal{S}_j$. We, thus, reformulate the problem as the maximization problem of log-likelihood ratio. If we do not assume any change point, i.e., $\mathcal{S}_0 = \emptyset$, Equation (3) is reduced to

$$L(\mathcal{D}; \hat{r}_1, \mathcal{S}_0) = -N - N \log \left(\frac{1}{N} \sum_{n=1}^{N} (t_n - t_{p(n)}) \right). \quad (4)$$

Thus, the log-likelihood ratio of the case where we assume J change points and the case where we assume no change points is given by

$$\begin{aligned} LR(\mathcal{S}_J) &= L(\mathcal{D}; \hat{\mathbf{r}}_{J+1}, \mathcal{S}_J) - L(\mathcal{D}; \hat{r}_1, \mathcal{S}_0) \\ &= N \log \left(\frac{1}{N} \sum_{n=1}^{N} (t_n - t_{p(n)}) \right) - \sum_{j=1}^{J+1} |\mathcal{D}_j| \log \left(\frac{1}{|\mathcal{D}_j|} \sum_{t_n \in \mathcal{D}_j} (t_n - t_{p(n)}) \right). \quad (5) \end{aligned}$$

We consider the problem of finding the set of change points $\mathcal{S}_J$ that maximizes $LR(\mathcal{S}_J)$ defined by Equation (5).

We note that, in general, it is conceivable that we are not able to acquire the complete tree structure of the diffusion sequence data. Thus, here, we consider two extreme cases, one in which the information spreads fastest (star shape tree) and the other in which the information spread slowest (line shape tree). The function which defines the parent node becomes $p(n) = 0$ for the former and $p(n) = n - 1$ for the latter. In case where there is no change point, the maximum likelihood estimator is $r^{-1} = (t_1 + \cdots + t_N)/N - t_0$ for the former and $r^{-1} = (t_N - t_0)/N$ for the latter. While we conjecture that in reality the optimal value lies in between these two extreme values, under the assumption that the actual tree structure of the diffusion data is unknown, we consider to approximate the optimal value by using either one of them. Here, note that in the former case, the maximum likelihood estimator represents the average diffusion delay time between the source node v_0 and each node v_i which is assumed to be connected to v_0 by a direct link, while in the latter case, it represents the average time interval between successive observation time points. Considering that the burst period we want to detect is much shorter than the other non burst periods, the latter case (line shape tree) seems to be more suitable for our aim. Therefore, $LR(\mathcal{S}_J)$ defined by Equation (5) becomes

$$LR(\mathcal{S}_J) = N \log \left(\frac{t_n - t_0}{N} \right) - \sum_{j=1}^{J+1} |\mathcal{D}_j| \log \left(\frac{T_j - T_{j-1}}{|\mathcal{D}_j|} \right). \quad (6)$$

We compared the bursts detected by using the two extreme values, and found that the use of line shape tree gave a better results and decided to use Equation (6) in our experiments.

4 Change Points Detection Method

We consider the problem of detecting change points as a problem of finding a subset $\mathcal{S}_J \subset \mathcal{D}$ when the set of time points of information diffusion result $\mathcal{D} = \{t_0, t_1, \cdots, t_N\}$ and the number of change points J are given. In other words, we search for J time points that are most likely to be the change points from a sequence of N observation points. In what follows, we explain each of the three methods, naive method (an exhaustive search), simple method (a greedy search), and the proposed method that is a combination of a greedy search and a local search.

4.1 Naive Method

The simplest method is to exhaustively search for the best set of J change points $\mathcal{S}_J$. Clearly the time complexity of this naive approach is $O(N^J)$. Thus, the number of change points detectable would be limited to $J = 2$ in order for the solution to be obtained in a reasonable amount of computation time when N is large enough.

4.2 Simple Method

We describe the simple method which is applicable when the number of change points J is large. This is a progressive binary splitting without backtracking. We fix the already selected set of $(j-1)$ change points $\mathcal{S}_{j-1}$ and search for the optimal j-th change point T_j and add it to $\mathcal{S}_{j-1}$. We repeat this procedure from $j = 1$ to J.

The algorithm is given below.

Step1. Initialize $j = 1$, $\mathcal{S}_0 = \emptyset$.
Step2. Search for $T_j = \arg\max_{t_n \in \mathcal{D}}\{LR(\mathcal{S}_{j-1} \cup \{t_n\})\}$.
Step3. Update $\mathcal{S}_j = \mathcal{S}_{j-1} \cup \{T_j\}$.
Step4. If $j = J$, output $\mathcal{S}_J$ and stop.
Step5. $j = j + 1$, and return to Step2.

Here note that in Step3 elements of the change point set $\mathcal{S}_j$ are reindexed to satisfy $T_{i-1} < T_i$ for $i = 2, \cdots j$. Clearly, the time complexity of the simple method is $O(NJ)$ which is fast. Thus, it is possible to obtain the result within a allowable computation time for a large N. However, since this is a greedy algorithm, it can be trapped easily to a poor local optimal.

4.3 Proposed Method

We propose a method which is computationally almost equivalent to the simple method but gives a solution of much better quality. We start with the solution obtained by the simple method $\mathcal{S}_J$, pick up a change point T_j from the already selected points, fix the rest $\mathcal{S}_J \setminus \{T_j\}$ and search for the better value T'_j of T_j, where $\cdot \setminus \cdot$ represents set difference. We repeat this from $j = 1$ to J. If no replacement is possible for all j $(j = 1, \cdots J)$, i.e. $T'_j = T_j$ for all j, no better solution is expected and the iteration stops.

The algorithm is given below.

Step1. Find $\mathcal{S}_J$ by the simple method and initialize $j = 1$, $k = 0$.
Step2. Search for $T'_j = \arg\max_{t_n \in \mathcal{D}}\{LR(\mathcal{S}_J \setminus \{T_j\} \cup \{t_n\})\}$.
Step3. If $T'_j = T_j$, set $k = k + 1$, otherwise set $k = 0$, and update $\mathcal{S}_J = \mathcal{S}_J \setminus \{T_j\} \cup \{T'_j\}$.
Step4. If $k = J$, output $\mathcal{S}_J$ and stop.
Step5. If $j = J$, set $j = 1$, otherwise set $j = j + 1$, and return to Step2.

It is evident that the proposed method requires computation time several times larger than that of the simple method, but it is much less than that of the naive method. How much the computation time increases compared to the simple method and how much the solution quality increases await for the experimental evaluation, which we will report in Section 5.

5 Experimental Evaluation

We experimentally evaluate the computation time and the accuracy of the change point detection using the real world Twitter information diffusion sequence data based on the methods we described in the previous section. We, then, analyze in depth the top 6 diffusion sequences in terms of the log-likelihood ratio based on the detected change points and burst periods, show that the line shape tree approximation is much better than the star shape tree approximation, and investigate whether or not we are able to identify which node in a social network caused the burst from the detected change points.

5.1 Experimental Settings

The information diffusion data we used for evaluation are extracted from 201,297,161 tweets of 1,088,040 Twitter users who tweeted at least 200 times during the three weeks from March 5 to 24, 2011 that includes March 11, the day of 2011 To-hoku earthquake and tsunami. It is conceivable to use a retweet sequence in which a user sends out other user's tweet without any modification. But there exists multiple styles of retweeting (official retweet and unofficial retweet), and it is very difficult to accurately extract a sequence of tweets in an automatic manner considering all of these different styles. Therefore, in our experiments, noting that each retweet includes the ID of the user who sent out the original tweet in the form of "@ID", we extracted tweets that include @ID format of each user ID and constructed a sequence data for each user. More precisely, we used information diffusion sequences of 798 users for which the length of sequences are more than 5,000 (number of tweets). Note that each diffusion sequence includes retweet sequences on multiple topics. Since we do not know the ground truth of the change points for each sequence if there are changes in it, we used the naive method which exhaustively search for all the possible combinations of the change points as giving the ground truth. We had to limit the number of change points to 2 ($J = 2$) in order for the naive method to return the solution in a reasonable amount of computation time. The experimental results explained in the next subsection is obtained by using a machine with Intel(R) Xeon(R) CPU W5590 @3.33GHz and 32GB memory.

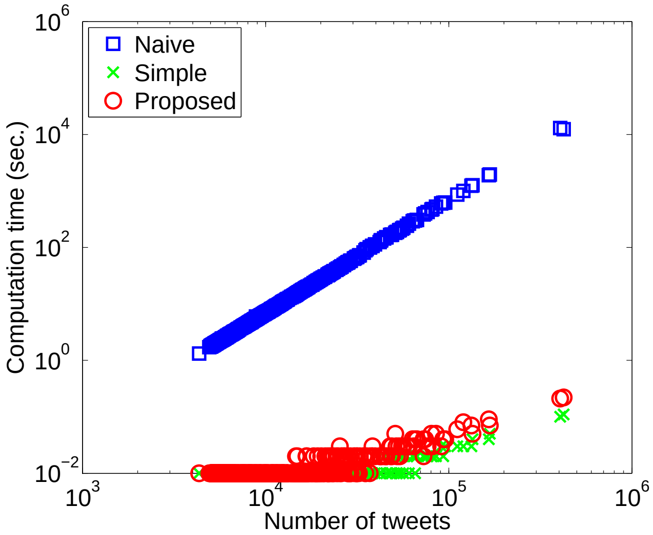


Fig. 1. Comparison of computation time among the three (naive, simple, and proposed) methods.

5.2 Main Results

Performance Evaluation Figure 1 shows the computation time that each method needed to produce the results. The horizontal axis is the length of the information diffusion data sequences, and the vertical axis is the computation time in second. The results clearly indicate that the naive method requires the largest computation time. The computation time is quadratic to the sequence length as predicted. In contrast, the computation time for the simple and the proposed methods is much shorter and it increases almost linearly to the increase of the sequence length for both. The proposed method requires more computation time due to the extra iteration needed for delayed backtracking. In fact, the number of extra iteration is 2.2 on the average and 7 at most.

Figure 2 shows the accuracy of the detected change points. We regarded that the solution obtained by the naive method is the ground truth. The horizontal axis is the sequence ranking of the log-likelihood ratio for the naive method (ranked from the top to the last), and the vertical axis is the logarithm of the likelihood ratio of the solution of each method. The results indicate that the simple method has lower likelihood ratio for all the range, meaning that it detects change points which are different from the optimal ones, but the proposed method can detect the correct optimal change points except for the low ranked sequences for which the likelihood ratio is small as is evident from the result in that the red curve representing the proposed method is indistinguishable from the blue curve representing the naive method. The reason why the accuracy of the proposed method for sequences with low likelihood decreases may be because the burst period is not clear for these sequences. In summary, out of the 798 sequences in total, the proposed method gave the correct results for 713 sequences (98.4%), whereas the simple method gave the correct results for only 171 sequences (21.4%). The average ratio of the likelihood ratio of the proposed method to that of the naive method (optimal

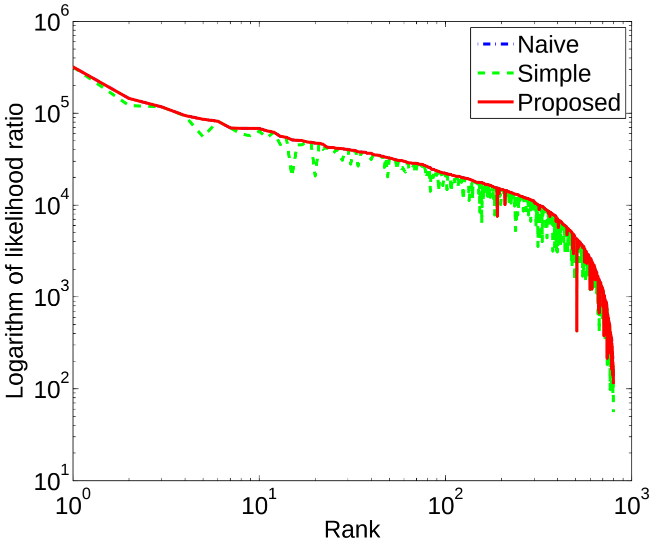


Fig. 2. Comparison of accuracy among the three (naive, simple, and proposed) methods.

solution) is 0.976, whereas the corresponding ratio for the simple method is 0.881, revealing that the proposed method gives much closer ratio to the optimal likelihood ratio. These results confirm that the proposed method can increase the change point detection accuracy to a great extent compared to the simple method with only a small penalty for the increased computation time.

In Depth Analyses on Detected Change Points and Burst Periods Next, we had a closer look at the top 6 diffusion sequences in terms of the log-likelihood ratios. Table 1 shows the total number of tweets included in the sequence, the starting and the ending time of the burst period, and the main topics that appeared near the beginning of the burst. Figure 3 shows how the cumulative number of tweets increases as time goes for each diffusion sequence. The horizontal axis is time and the vertical axis is the cumulative number of tweets. The two red vertical lines in the graph are the change (starting and ending) points detected by the proposed method, and the interval between them is the burst period.

As is understood from Table 1, explosive retweeting of the information of urgent need about the earthquake for a short period of time triggered the start of the burst (with the exception of the 4th ranked sequence). The 4th ranked sequence is for the account called “ordinary timeline” which was set up for allowing to tweet everyday topics by adding “@itsumonoTL” at the beginning of the tweet when people are in voluntary restraint mood after the disastrous earthquake. We can say, with the exception

[1] NHK is the government operated broadcaster.

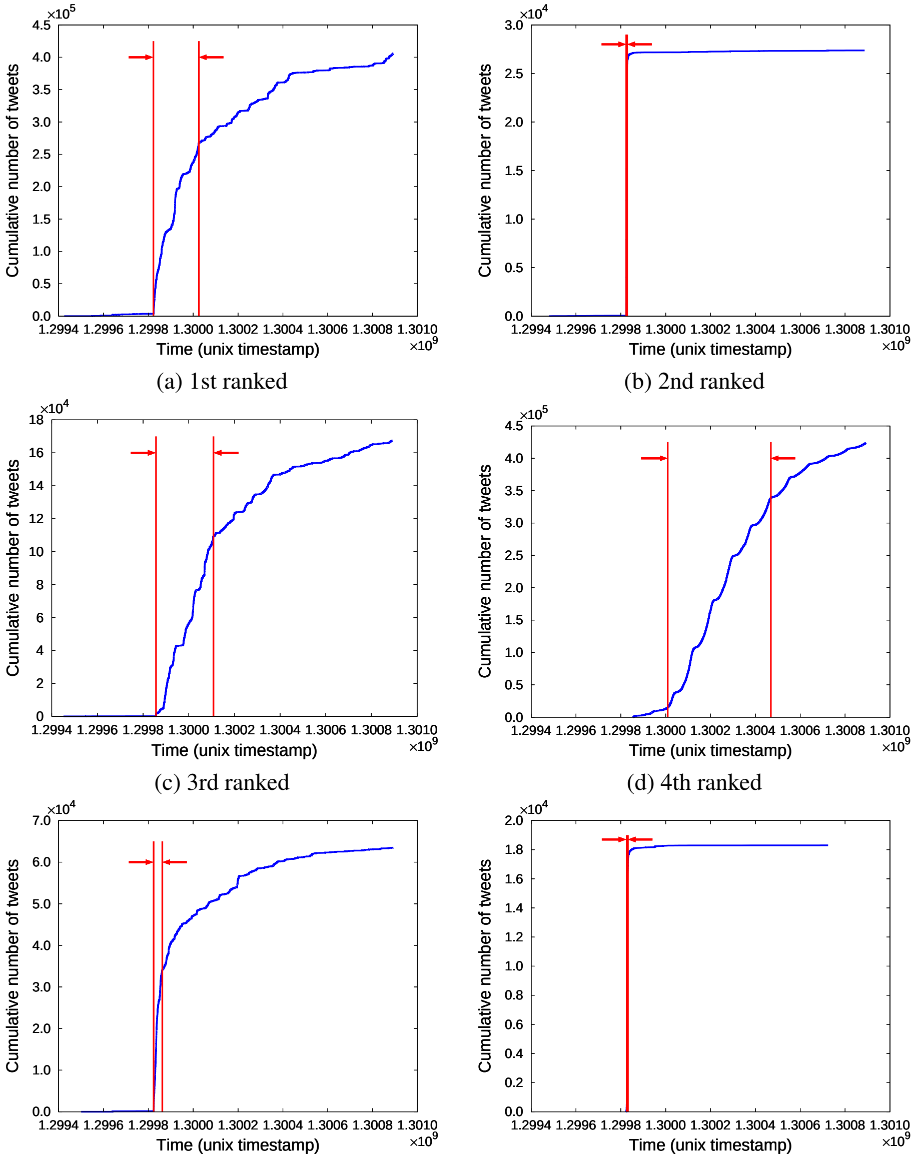
[2] Great Hanshin-Awaji Earthquake occurred on January 17, 1995 in Kobe area and 6,434 people lost their lives.

Table 1. Major topics appearing at the beginning of the burst periods of the top 6 diffusion results in terms of log-likelihood ratio

Ranking	Length	Detected burst period		Major topics at the beginning of the burst period
		Start	End	
1	450,739	2011/3/11 14:48:13	2011/3/13 23:13:04	Retweets of the earthquake bulletin posted by the PR department of Japan Broadcasting Corporation, NHK (@NHK_PR).[1]
2	27,372	2011/3/11 15:13:57	2011/3/11 16:19:26	Retweets of the article on to-do list at the time of earthquake onset posted by a victim of the Great Hanshin-Awaji Earthquake. [2]
3	167,528	2011/3/12 00:18:19	2011/3/14 22:08:20	Retweets of the article on measures against cold at an evacuation site posted by the news department of NHK (@nhk_seikatsu).
4	423,594	2011/3/13 18:38:50	2011/3/19 02:20:58	Ordinary tweets irrelevant to the earthquake posted to a special account "@itsumonoTL".
5	63,485	2011/03/11 15:05:08	2011/03/12 01:52:13	Retweets of the earthquake bulletin posted by the Fire and Disaster Management Agency (@FDMA_JAPAN).
6	18,299	2011/3/11 15:45:17	2011/3/11 17:19:02	Retweets of a call for help posted by a user who seemed to be buried under a server rack (later found to be a false rumor).

of such a special case of "ordinary timeline", that we are able to detect efficiently a time period where tweets on a specific topic (of urgent need in this example) are intensively retweeted by looking at the change points detected by the proposed method even from the diffusion sequence that contains multiple topics.

We note that the cumulative number of the tweets for the 2nd and 6th ranked diffusion sequences is smaller than the other 4 sequences from Table 1, and the burst period of these 2 sequences are much shorter than others and there is little changes in the number of tweets before and after the burst from Figure 3. This difference is considered to come from whether the account is private or public. Among these 4 sequences, except for the exceptional 4th one, the remaining 3 are all from the public organization accounts (1st and 3rd are NHK and 5th is FDMA). Information posted by these accounts tends to disseminate widely everyday. Thus, considering this situation, it is natural to observe that the cumulative number of tweets shows a relatively smooth increase as seen in Figure 3 by adding multiple bursts of short periods about the earthquake-related information of urgent need as shown in Table 1. Figure 3(e) has only one smooth change during the burst period, which indicates that the earthquake bulletin in Table 1 is the only source of the burst. On the other hand, we see multiple smooth changes with discontinuity of the gradient at each boundary during the burst period in Figures 3(a) and (c). This implies that there can be other sources of the burst than shown in Table 1. Indeed, it is possible to identify these change points by increasing the value of J (an example explained later). On the other hand, Figures 3(b) and (f) shows that the information posted by an individual that is rarely retweeted in ordinary situations can be propagated explosively if it is of urgent need, e.g. timely information about earthquake.



(a) 1st ranked

(b) 2nd ranked

(c) 3rd ranked

(d) 4th ranked

(e) 5th ranked

(f) 6th ranked

Fig. 3. Temporal change of cumulative number of tweets in the top 6 diffusion results in terms of the highest log-likelihood ratio

Here, we report the result when we increase the number of change points. Figure 4 shows the result for the 3rd ranked sequence in Figure 3(c) when J is set to 9. There are 9 vertical lines corresponding to each change point, but the first two change points are too close and indistinguishable. Note that horizontal axis is enlarged and the range

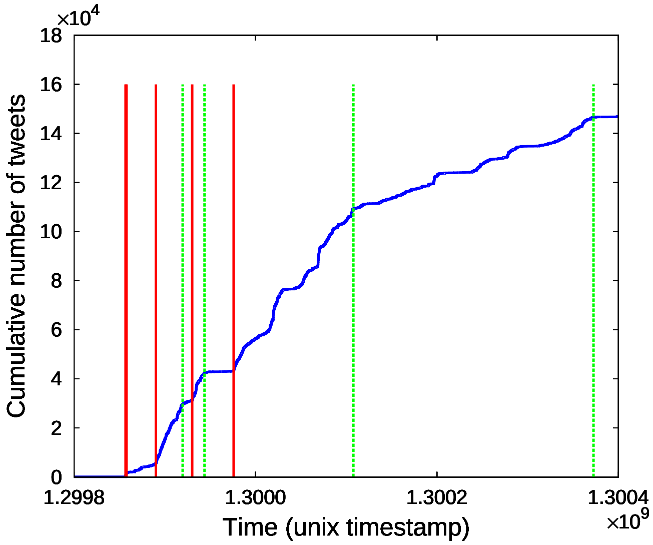
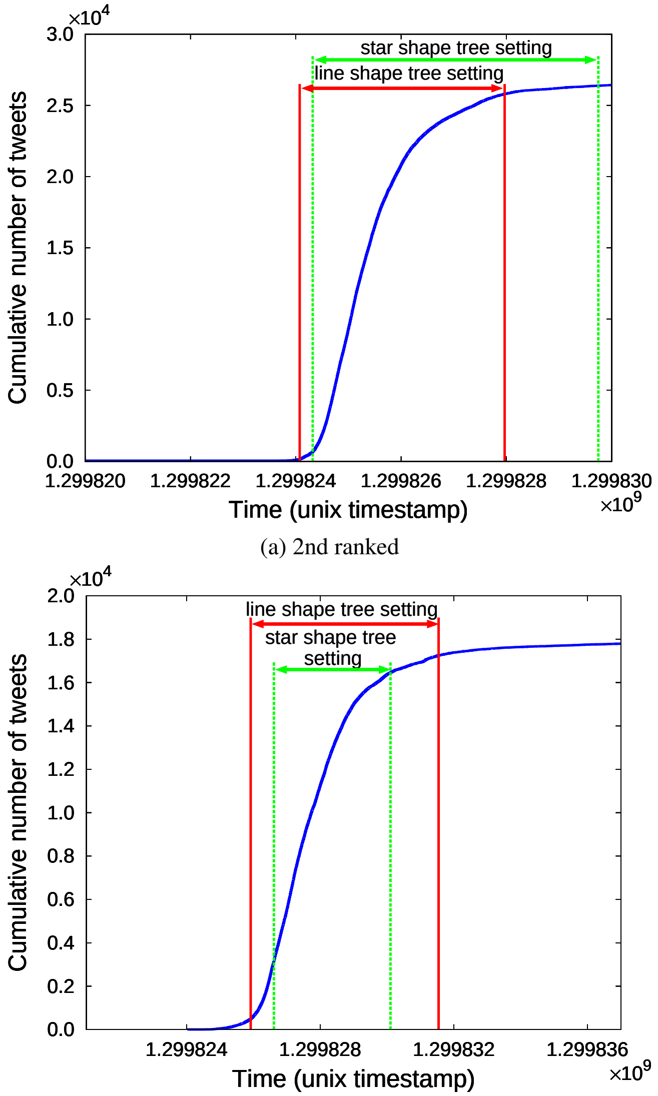


Fig. 4. Finer burst detection for the 3rd ranked sequence in Figure 4(c) when J is set to 9

shown is different from that in Figure 3(c). We see that the detected change points are located at the boundary points where the gradients of the curves change discontinuously. Those 4 broken lines in green are considered to indicate the end of the burst because the gradient change across each boundary is rather smaller. In fact, we investigated the most recent 10 tweets for these 4 change points and confirmed that no more than half of the retweets is talking about the same topic except the one second from the last in which 7 of them are on the same topic. The remaining 5 change points (red lines) all contain at least 7 retweets (10, 8, 7, 7, 9) that are on the same topic. From this fact, we can reconfirm that there appear many tweets on the same topic during the burst period.

Line Shape Tree vs. Star Shape Tree Note that all of these results were obtained by assuming that the information diffuses along the line shape tree as discussed in Section 3. Here, we show that use of line shape tree gives better results than use of star shaped tree. To this end, we compared the bursts detected for the 2nd and 6th ranked information diffusion sequences which include only one burst.

The results are illustrated in Figure 5, where red solid and green broken vertical lines denote the change points detected by the naive method with the line shape and star shape settings, respectively. Only the time range of interest is extracted and shown in the horizontal axis. From these figures, we observe that use of line shape tree detects the change points more precisely as expected, which means that line shape tree gives a better approximation of the maximum likelihood estimator than star shape tree even if the actual tree shape of the diffusion path is not known to us.



(a) 2nd ranked

(b) 6th ranked

Fig. 5. Comparison of bursts detected by use of line shape tree and star shape tree for the 2nd and 6th ranked information diffusion sequences in Table 1.

Change Points in a Time Line and Nodes in a Network Remember that each observed time point corresponds to a node in a social network. In this sense, it can be said that the proposed method detects not only the change points in a time line, but also the change points in a network. However, unfortunately, those nodes do not necessarily correspond to those which actually caused the burst period. For example, in the second ranked sequence in Table 1, we observed at least 1 retweet of the article described in

Table 1 per second after the start of the burst, 2011/3/11 15:13:57, while we observed at most 20 per minute before the burst started. This shows the accuracy of the detected change point, but it also means that the node that actually influenced nodes within the burst period could exist in the period before the change point. Indeed, we observed the first retweet at 2011/3/11 15:07:05 and 69 retweets thereafter before the change point. It is natural to think that some of them played an important role on the explosive diffusion of the article. We need to know the actual information diffusion path to find such important nodes, but detecting change points in a time line would significantly reduce the effort needed to do so because the search can be focused on the limited sub-sequences around the change points. Devising a method to find such important nodes is one of our future work.

6 Conclusion

We addressed the problem of detecting the period in which information diffusion burst occurs from a single observed diffusion sequence under the assumption that the delay of the information propagation over a social network follows the exponential distribution. To be more precise, we formulated the problem of detecting the change points and finding the values of the time delay parameter in the exponential distribution as an optimization problem of maximizing the likelihood of generating the observed diffusion sequence. We devised an efficient iterative search algorithm for the change point detection whose time complexity is almost linear to the number of data points. We tested the algorithm against the real Twitter data of the 2011 To-hoku earthquake and tsunami, and experimentally confirmed that the algorithm is much more efficient than the exhaustive naive search and is much more accurate than the simple greedy search. By analyzing the real information diffusion data, we revealed that even if the data contains tweets talking about plural topics, the detected burst period tends to contain tweets on a specific topic intensively. In addition, we experimentally confirmed that assuming the information diffusion path to be the line shape tree results in much better approximation of the maximum likelihood estimator than assuming it to be the star shape tree. This is a good heuristic to accurately estimate the change points when the actual diffusion path is not known to us. These results indicate that it is possible to detect and identify both the burst period and the topic diffused without extracting the tweet sequence for each topic and identifying the diffusion paths for each sequence, and the proposed method can be a useful tool to analyze a huge amount of information diffusion data. Our immediate future work is to compare the proposed method with existing burst detection methods that are designed for data stream. We also plan to devise a method of finding nodes that caused the bust based on the change points detected.

Acknowledgments

The Tweeter data we used in this paper were provided by Prof. Fujio Toriumi of Tokyo University and Dr. Kazuhiro Kazama of Nippon Telegraph and Telephone Corporation.

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-114111, and JSPS Grant-in-Aid for Scientific Research (C) (No. 23500194).

References

1. Bonacichi, P.: Power and centrality: A family of measures. Amer. J. Sociol. 92, 1170–1182 (1987)
2. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters 12, 211–223 (2001)
3. Katz, L.: A new status index derived from sociometric analysis. Sociometry 18, 39–43 (1953)
4. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003). pp. 137–146 (2003)
5. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Min. Knowl. Disc. 20, 70–97 (2010)
6. Kleinberg, J.: Bursty and hierarchical structure in streams. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002). pp. 91–101 (2002)
7. Sadikov, E., Medina, M., Leskovec, J., Garcia-Molina, H.: Correcting for missing data in information cascades. In: Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM 2011). pp. 55–64 (2011)
8. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Learning continuous-time information diffusion model for social behavioral data analysis. In: Proceedings of the 1st Asian Conference on Machine Learning (ACML2009). pp. 322–337. LNAI 5828 (2009)
9. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Selecting information diffusion models over social networks for behavioral analysis. In: Proceedings of the 2010 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2010). pp. 180–195. LNAI 6323 (2010)
10. Sun, A., Zeng, D., Chen, H.: Burst detection from multiple data streams: A network-based approach. IEEE Transactions on Systems, Man, & Cybernetics Society, Part C pp. 258–267 (2010)
11. Wasserman, S., Faust, K.: Social network analysis. Cambridge University Press, Cambridge, UK (1994)
12. Watts, D.J.: A simple model of global cascades on random networks. Proceedings of National Academy of Science, USA 99, 5766–5771 (2002)
13. Watts, D.J., Dodds, P.S.: Influence, networks, and public opinion formation. Journal of Consumer Research 34, 441–458 (2007)
14. Zhu, Y., Shasha, D.: Efficient elastic burst detection in data streams. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003). pp. 336–345 (2003)

Graph Embedding on Spheres and its Application to Visualization of Information Diffusion Data

Kazumi Saito
University of Shizuoka
Shizuoka, Japan
k-saito@u-shizuoka-ken.ac.jp

Masahiro Kimura
Ryukoku University
Otsu, Japan
kimura@rins.ryukoku.ac.jp

Kouzou Ohara
Aoyama Gakuin University
Kanagawa, Japan
ohara@it.aoyama.ac.jp

Hiroshi Motoda
Osaka University
Osaka, Japan
motoda@ar.sanken.osaka-u.ac.jp

ABSTRACT

We address the problem of visualizing structure of undirected graphs that have a value associated with each node into a K-dimensional Euclidean space in such a way that 1) the length of the point vector in this space is equal to the value assigned to the node and 2) nodes that are connected are placed as close as possible to each other in the space and nodes not connected are placed as far apart as possible from each other. The problem is reduced to K-dimensional spherical embedding with a proper objective function. The existing spherical embedding method can handle only a bipartite graph and cannot be used for this purpose. The other graph embedding methods, e.g., multi-dimensional scaling, spring force embedding methods, etc., cannot handle the value constraint and thus are not applicable, either. We propose a very efficient algorithm based on a power iteration that employs the double-centering operations. We apply the method to visualize the information diffusion process over a social network by assigning the node activation time to the node value, and compare the results with the other visualization methods. The results applied to four real world networks indicate that the proposed method can visualize the diffusion dynamics which the other methods cannot and the role of important nodes, e.g. mediator, more naturally than the other methods.

Categories and Subject Descriptors

I.2.6 [**Learning**]: Parameter learning

Keywords

graph embedding, visualization, information diffusion

1. INTRODUCTION

Complex network is hard to understand. Visualization can help, but in reality it is not self-evident whether there exists a good general visualization scheme that satisfies most of our needs. Especially if we want to visualize the dynamics taking place over a network, the only solution seems to use animation over time, which is not what we are aiming at.

We consider the following problem: Visualize the structure of an undirected graph that has a value assigned to each node in a K-dimensional Euclidean space in such a way that 1) the length of the point vector in this space is equal to the node value and 2) nodes that are connected are placed as close as possible to each other in the space and nodes not connected are placed as far apart as possible from each other. The constraint 1) is unique to this method and brings more flexibility in visualization. In fact, this enables to visualize a dynamics mentioned in the beginning.

The need for visualization is so high that various graph embedding methods have already been proposed and are widely used. These include multi-dimensional scaling [15], spectral embedding [2], spring force embedding [4] and cross-entropy embedding [16]. All of them are applicable to undirected graphs. Spherical embedding [11, 3] that came a little later is designed to visualize bipartite graphs. Among these five, the first four cannot handle the constraint 1). The last one cannot apply to a general undirected graph. To our knowledge, there is no method that can directly handle our problem. Further, apart from the above problem, those that solve non-linear optimization problem by a power iteration, except [3], are extremely slow.

We show that the above visualization problem is reduced to spherical embedding that is formulated as a non-linear optimization problem which maximizes a certain objective function that involves an operation called "double-centering". The problem can be solved by a simple power iteration as is done in the above existing methods, but this is very inefficient. We propose a much more efficient algorithm making effective use of the sparsity of the adjacency matrix, which is true for most complex networks. We verify that the algorithm works as intended by applying it to the visualization of information diffusion process over a large social network by assigning the node activation time to the node value (detail in Section 4.2). The results obtained by four real world social networks confirm our conjecture. Time evolution of the diffusion process is easily visualized by the proposed method and in this process such nodes that have a role of mediating the diffusion are more easily identifiable than the other existing methods which cannot handle the diffusion dynamics.

This paper is organized as follows. We first describe the

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

problem framework of embedding undirected graphs into a low dimensional Euclidean space (2.1.), show a simple update method for solving the optimal solution (2.1), followed by the proposed efficient update method (2.3). Next we briefly compare the proposed method with four existing methods (3). We then explain how we apply the method to the visualization of information diffusion (4), and report the results (5). We conclude the paper by summarizing what has been achieved (7).

2. SPHERICAL GRAPH EMBEDDING

We describe the framework of embedding an undirectded graph $G = (V, E)$ without self-loops into a K-dimensional Euclidean space, where V and E ($\subset V \times V$) stand for the sets of all the nodes and links, respectively. For the sake of technical convenience, we identify the set of the nodes, V, by a series of positive integers, i.e., $V = \{1, \cdots, m, \cdots, M\}$. Here M is the number of the nodes in V , i.e., $|V| = M$. Then, we can define the $M \times M$ adjacency matrix $\mathbf{A} = \{a_{m,n}\}$ by setting $a_{m,n} = 1$ if $\{m, n\} \in E$; $a_{m,n} = 0$ otherwise. Note taht $a_{m,n} = a_{n,m}$ and $a_{m,m} = 0$. We denote the K-dimensional embedding position vectors by $\mathbf{x}_m$ for the node $m \in V$. Then we can construct the $K \times M$ matrix consisting of these position vectors, i.e., $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_M)$.

2.1 Problem Formulation

We first state the framework of our embedding problem intuitively: For a given undirectded graph $G = (V, E)$ and a set of values assigned to each node, denoted by $(r_1, \cdots, r_m, \cdots, r_M)$, we attempt to visualize the graph so that each pair of nodes with similar connection patterns is embedded as a pair of position vectors with similar directions, and each length of the embedded position vectors is set to the above value assigned to the node, i.e., $\|\mathbf{x}_m\| = r_m$ for each m, where $\|\mathbf{x}_m\|$ stands for the norm of the vector $\mathbf{x}_m$.

In order to more closely explain our embedding problem, we introduce the centering (Young-Householder transformation) matrix,

$$\mathbf{H}_M = \mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T, \quad (1)$$

where $\mathbf{I}_M$ stands for the $M \times M$ identity matrix, $\mathbf{1}_M$ is an M-dimensional vector whose elements are all one, and $\mathbf{1}^T$ means the transposition of the vector $\mathbf{1}$. Clearly, the mean vector of the resulting position vectors becomes $\mathbf{0}$ by the operations $\mathbf{X}\mathbf{H}_M$. Then, we consider the following double-centered matrix $\mathbf{B} = \{b_{m,n}\}$ that is calculated from the adjacency matrix $\mathbf{A}$.

$$\mathbf{B} = \mathbf{H}_M\mathbf{A}\mathbf{H}_M. \quad (2)$$

Note that the mean vectors of both the row and the column vectors of the matrix $\mathbf{B}$ become $\mathbf{0}$. On the other hand, for position vectors $\{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$, we can consider the similarity matrix $\mathbf{C} = \{c_{m,n}\}$, each element of which is defined by the following cosine similarity.

$$c_{m,n} = \frac{\mathbf{x}_m^T}{\|\mathbf{x}_m\|}\frac{\mathbf{x}_n}{\|\mathbf{x}_n\|}. \quad (3)$$

As the basic strategy of our graph embedding, we maximize the correlation between the the double-centered matrix $\mathbf{B}$ and the cosine similarity matrix $\mathbf{C}$ by adequately locating each position vector under the constraints $\|\mathbf{x}_m\| = r_m$. Namely, we can consider the following objective function with respect to the matrix $\mathbf{X}$ constructed from the position vectors.

$$\begin{aligned} J(\mathbf{X}) &= \sum_{m=1}^{M-1}\sum_{n=m+1}^{M} b_{m,n}c_{m,n} + \frac{1}{2}\sum_{m=1}^{M}\lambda_m(r_m^2 - \mathbf{x}_m^T\mathbf{x}_m) \\ &= \sum_{m=1}^{M-1}\sum_{n=m+1}^{M} b_{m,n}\frac{\mathbf{x}_m^T}{r_m}\frac{\mathbf{x}_n}{r_n} + \frac{1}{2}\sum_{m=1}^{M}\lambda_m(r_m^2 - \mathbf{x}_m^T\mathbf{x}_m) \end{aligned} \quad (4)$$

where $\{\lambda_m \mid m = 1, \cdots, M\}$ correspond to Lagrange multipliers for the constraints, i.e., $\mathbf{x}_m^T\mathbf{x}_m = r_m^2$ for $1 \le m \le M$. Intuitively, maximizing $J(\mathbf{X})$ pushes the pairs $\mathbf{x}_m$ and $\mathbf{x}_n$ to the same direction if they are connected and pushes them to the opposite direction if they are unconnected, and realizes the intended visualization.

Now, we consider a reparameterization of each position vector $\mathbf{x}_m$ by $\tilde{\mathbf{x}}_m = \mathbf{x}_m/r_m$, and set $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_M)^T$. Then, we can equivalently transform our objective function defined in Equation (4) as follows,

$$J(\tilde{\mathbf{X}}) = \sum_{m=1}^{M-1}\sum_{n=m+1}^{M} b_{m,n}\tilde{\mathbf{x}}_m^T\tilde{\mathbf{x}}_n + \frac{1}{2}\sum_{m=1}^{M}\mu_m(1 - \tilde{\mathbf{x}}_m^T\tilde{\mathbf{x}}_m), \quad (5)$$

where $\mu_m = \lambda_m/r_m^2$ for each m. Thus, maximizing Equation (4) is implemented by the following two steps: First, we calculate the position vector $\tilde{\mathbf{x}}_m$ for each node on the unit sphere (circle), so as to maximize Equation (5); Then, we can obtain the final position vectors just by rescaling them with respect to $(r_1, \cdots, r_M)$, i.e., $\mathbf{x}_m = r_m\tilde{\mathbf{x}}_m$ for each m. Thus we can regard our problem as a shperical graph embedding problem on the unit sphere. Hereafter, we simply denote $\tilde{\mathbf{x}}_m$ as $\mathbf{x}_m$ in order to avoid notational complication. Here we should emphasize that in our problem formalization, the directions of the embedded position vectors are determined independently from the values assigned to each node.

2.2 Simple Update Method

Now we consider maximizing $J(\mathbf{X})$ defined in Equation (5) by use of a coordinate strategy: We maximize $J(\mathbf{X})$ with respect to each position vector $\mathbf{x}_m$, by fixing the other position vectors. In order to optimally update each position vector $\mathbf{x}_m$, we consider the following gradient vector of the objective function $J(\mathbf{X})$ with respect to $\mathbf{x}_m$.

$$\frac{\partial J(\mathbf{X})}{\partial \mathbf{x}_m} = \sum_{n=1, n\neq m}^{M} b_{m,n}\mathbf{x}_n - \mu_m\mathbf{x}_m. \quad (6)$$

Thus, for the fixed vectors $\{\mathbf{x}_1, \cdots \mathbf{x}_M\} \setminus \mathbf{x}_m$, we obtain the optimal position vector $\mathbf{x}_m$ which maximizes the objective function $J(\mathbf{X})$ as follow:

$$\mathbf{x}_m = \frac{1}{\|\mathbf{f}_m\|}\mathbf{f}_m, \quad (7)$$

where

$$\mathbf{f}_m = \sum_{n=1, n\neq m}^{M} b_{m,n}\mathbf{x}_n = (\mathbf{X} - \mathbf{x}_m\mathbf{e}_m^T)\mathbf{B}\mathbf{e}_m. \quad (8)$$

Here $\mathbf{e}_m$ is an M-dimensional unit vector whose m-th element is 1, and the other elements are 0.

However, this simple iteration method requires the computational complexity of $O(MK)$ for updating each optimal position vector according to Equation (8). In order to make better use of the sparsity of adjacency matrix which is frequently observed in most complex networks, we derive an efficient way of calculating Equation (8) in the succeeding subsection.

2.3 Efficient Update Method

We first focus on the following equivalent formula for calculating $\mathbf{f}_m$ in Equation (8).

$$\mathbf{f}_m = \mathbf{X}\mathbf{B}\mathbf{e}_m - (\mathbf{e}_m^T \mathbf{B}\mathbf{e}_m)\mathbf{x}_m. \tag{9}$$

Here we consider a degree vector defined by

$$\mathbf{d} = (d_1, \cdots, d_M)^T = \mathbf{A}\mathbf{1}_M, \tag{10}$$

and their average,

$$D = \frac{1}{M}\mathbf{1}_M^T \mathbf{d} = \frac{1}{M}\mathbf{1}_M^T \mathbf{A}\mathbf{1}_M. \tag{11}$$

Then, from the definition of double-centered matrix $\mathbf{B}$ given in Equation (2), we can calculate $\mathbf{B}\mathbf{e}_m$ as follows.

$$\begin{aligned} \mathbf{B}\mathbf{e}_m &= (\mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T)\mathbf{A}(\mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T)\mathbf{e}_m \\ &= \mathbf{A}\mathbf{e}_m + \frac{D - d_m}{M}\mathbf{1}_M - \frac{1}{M}\mathbf{d}. \end{aligned} \tag{12}$$

By noting that $\mathbf{e}_m^T \mathbf{A}\mathbf{e}_m = 0$ because of no self-loops, we obtain $\mathbf{e}_m^T \mathbf{B}\mathbf{e}_m$ as follows.

$$\mathbf{e}_m^T \mathbf{B}\mathbf{e}_m = \frac{D - 2d_m}{M} \tag{13}$$

Now we define the average position vector $\boldsymbol{\phi}$ and the degree-weighted average position vector $\boldsymbol{\psi}$ by

$$\boldsymbol{\phi} = \frac{1}{M}\mathbf{X}\mathbf{1}_M, \quad \boldsymbol{\psi} = \frac{1}{M}\mathbf{X}\mathbf{d}, \tag{14}$$

respectively. Then by substituting Equations (12) and (13) into Equation (9), we can obtain the following.

$$\mathbf{f}_m = \sum_{n \in \Gamma(m)} \mathbf{x}_n + (D - d_m)\boldsymbol{\phi} - \boldsymbol{\psi} - \frac{D - 2d_m}{M}\mathbf{x}_m, \tag{15}$$

where, $\Gamma(m)$ denotes a set of neighbour nodes of v, *i.e.*, those nodes that are connected to v. Thus by noting that both $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are K-dimensional vectors, and the average number of elements in $\Gamma(m)$ is D, i.e., $D =< |\Gamma(m)| >$, we can see that the average computational complexity of calculating $\mathbf{f}_m$ is reduced to $O(DK)$ from $O(MK)$ in average. As mentioned earlier, we can naturally assume $M \gg D$ for a wide variety of complex networks.

On the other hand, after updating the position vector $\mathbf{x}_m$, we need to update vectors $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ according to this change as well. For this purpose, after setting the updated vector $\mathbf{y}_m$ and the modification vector $\Delta\mathbf{x}_m$ by,

$$\mathbf{y}_m = \frac{1}{\|\mathbf{f}_m\|}\mathbf{f}_m, \quad \Delta\mathbf{x}_m = \mathbf{y}_m - \mathbf{x}_m, \tag{16}$$

we update the vectors $\boldsymbol{\phi}$, $\boldsymbol{\psi}$, and $\mathbf{x}_m$ as follows.

$$\boldsymbol{\phi} = \boldsymbol{\phi} + \frac{1}{M}\Delta\mathbf{x}_m, \quad \boldsymbol{\psi} = \boldsymbol{\psi} + \frac{d_m}{M}\Delta\mathbf{x}_m, \quad \mathbf{x}_m = \mathbf{y}_m. \tag{17}$$

Clearly, these updates can be done within the computational complexity of $O(K)$. Thus, we can see that the computational complexity of updating $\mathbf{x}_m$ is equal to $O(DK)$.

Below we summarize our spherical embedding algorithm proposed in this paper.

1. Initialize position vectors $\{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$ adequately; and calculate vectors $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ by Equation (14);

2. For each $m \in \{1, \cdots, M\}$, calculate $\mathbf{f}_m$ by Equation (15), set vectors $\mathbf{y}_m$ and $\Delta\mathbf{x}_m$ by Equation (16), and update vectors $\boldsymbol{\phi}$, $\boldsymbol{\psi}$, and $\mathbf{x}_m$ by Equation (17);

3. If $\max_m\{\|\partial J(\mathbf{X})/\partial \mathbf{x}_m\|\} < \epsilon$, output $\{\mathbf{x}_1, \cdots, \mathbf{x}_M\}$ and terminate;

4. Return to the step 2.

Our proposed algorithm employs a power iteration as the basic framework, just like the HITS algorithm [8], which utilizes $\mathbf{A}$ and $\mathbf{A}^T$, does. However, the main differences are use of the double-centering operation by $\mathbf{H}_M$, and the constraints described by $\|\mathbf{x}_m\| = r_m$. Here note that the double-centering operation is also employed in the standard multidimensional scaling method [15].

Now we briefly mention the computational complexity of our algorithm. Clearly, the main computational complexity of one-iteration comes from the multiplication by the matrix $\mathbf{A}$ with position vectors $\mathbf{x}_m$, which is the most computationally intensive part and is proportional to the number of links in the undirected graph. Thus, the proposed algorithm is expected to work much faster especially for a sparse undirected graph. In fact, it has been well known that the PageRank algorithm [1] based on a power iteration works very fast for a large and sparse network [10] even without parallel distributed processing.

3. ALGORITHMIC COMPARISON WITH CONVENTIONAL METHODS

We compare the proposed method from algorithmic aspect with the four well known embedding methods: multidimensional scaling [15], spectral embedding [2], spring force embedding [4], and cross-entropy embedding [16]. Here the former two perform a power iteration with respect to either a double-centered distance matrix or a graph Laplacian matrix which is calculated from a given graph, while the latter two repeatedly move each position vector by using the Newton method in a framework of nonlinear optimization. Here note that the basic strategy of our method is a combination of the above basic strategy, i.e., our method performs a power iteration with respect to a double-centered adjacency matrix while repeatedly moving each position vector. However, recall that these existing methods cannot directly utilize the values associated with nodes. In what follows, we compare our method more closely with these existing methods.

Multi-dimensional scaling method [15] first calculates the distance matrix $\mathbf{G}$, and performs the double centering operation to the distance matrix. Mathematically it is formulated as minimizing Equation (18).

$$\mathcal{M}(\mathbf{X}) = \frac{1}{2}\sum_{k=1}^{K} \mathbf{z}_k^T (\mathbf{H}_M \mathbf{G} \mathbf{H}_M)\mathbf{z}_k, \tag{18}$$

where $\mathbf{z}_k = (x_{1,k}, \cdots, x_{M,k})^T$, and $\{\mathbf{z}_1, \cdots, \mathbf{z}_K\}$ need to be orthonormal vectors, i.e., $\mathbf{z}_k^T \mathbf{z}_k = 1$ and $\mathbf{z}_k^T \mathbf{z}_{k'} = 0$ if $k \neq k'$.

Spectral embedding method [2] tries to directly minimize distances between position vectors of connecting nodes. Mathematically it is formulated as minimizing Equation (19).

$$\begin{aligned}\mathcal{S}(\mathbf{X}) &= \frac{1}{2}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{n=1}^{N} a_{m,n}(z_{k,m}-z_{k,n})^2 \\ &= \sum_{k=1}^{K}\mathbf{z}_k^T(\mathbf{D}-\mathbf{A})\mathbf{z}_k, \qquad (19)\end{aligned}$$

where $\mathbf{D}$ is a diagonal matrix each element of which is the degree of node (number of links). Note that $(\mathbf{D}-\mathbf{A})$ is referred to as a graph Laplacian matrix. Again, we set $\mathbf{z}_k = (x_{1,k}, \cdots, x_{M,k})^T$, and $\{\mathbf{z}_1, \cdots, \mathbf{z}_K\}$ need to be orthonormal vectors, which excludes the trivial vector expressed as $\mathbf{z} \propto \mathbf{1}_M$.

Spring force embedding method [4] assumes that there is a hypothetical spring between each connected node pair and locates nodes such that the distance of each node pair is closest to its minimum path length at equilibrium. Mathematically it is formulated as minimizing Equation (20).

$$\mathcal{K}(\mathbf{X}) = \sum_{m=1}^{M-1}\sum_{n=m+1}^{M}\alpha_{m,n}(g_{m,n} - \|\mathbf{x}_m - \mathbf{x}_n\|)^2, \qquad (20)$$

where $\alpha_{m,n}$ is a spring constant which is normally set to $1/(2g_{u,v}^2)$.

Cross-entropy embedding method [16] first defines a similarity $\rho(\mathbf{x}_m, \mathbf{x}_n)$ between the embedding positions x_m and x_n and uses the corresponding element $a_{m,n}$ of the adjacency matrix as a measure of distance between the node pair, and tries to minimize the total cross entropy between these two. Mathematically it is formulated as minimizing Equation (21).

$$\begin{aligned}\mathcal{C}(\mathbf{X}) &= -\sum_{m=1}^{M-1}\sum_{n=m+1}^{M}(a_{m,n}\log\rho(\mathbf{x}_m, \mathbf{x}_n) \\ &\quad +(1-a_{m,n})\log(1-\rho(\mathbf{x}_u, \mathbf{x}_v))). \qquad (21)\end{aligned}$$

Here, note that we used the function $\rho(\mathbf{x}_u, \mathbf{x}_v) = \exp(-\frac{1}{2}||\mathbf{x}_u - \mathbf{x}_v||^2)$ in our experiments.

The spectral embedding method is expected to work comparable to our method because these methods perform a power iteration on a sparse adjacency matrix. The multidimensional scaling method requires a substantially large computation time because it needs to perform a power iteration on a full distance matrix. Spring force embedding method and Cross-entropy embedding method both of which repeatedly move each position vector by using the Newton method, require an extremely large computation time before the final results are obtained.

4. APPLICATION TO VISUALIZATION OF INFORMATION DIFFUSION DATA

Our primary application of the proposed method is visualization of information diffusion process over a social network. We start with a brief description of the diffusion models we used and then describe how we visualize the diffusion data.

4.1 Information diffusion models

We focus on the IC (Independent Cascade) and the LT (Linear Threshold) models [5] as the representative models of information diffusion, and utilize their extended version that can cope with asynchronous time activation, AsIC (Asynchronous IC) and AsLT (Asynchronous LT) models [13, 14] in our experiments.

4.1.1 Asynchronous Independent Cascade Model

We first recall the definition of the IC model according to the work of [5], and then introduce the AsIC model. In the IC model, we specify a real value $p_{m,n}$ with $0 < p_{m,n} < 1$ for each link (m, n) in advance. Here $p_{m,n}$ is referred to as the *diffusion probability* through link (m, n). The diffusion process unfolds in discrete time-steps $t \geq 0$, and proceeds from a given initial active set S in the following way. When a node m becomes active at time-step t, it is given a single chance to activate each currently inactive child node n, and succeeds with probability $p_{m,n}$. If m succeeds, then n will become active at time-step $t+1$. If multiple parent nodes of n become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not m succeeds, it cannot make any further attempts to activate n in subsequent rounds. The process terminates if no more activations are possible.

In the AsIC model, we specify real values $r_{m,n}$ with $r_{m,n} > 0$ in advance for each link $(m, n) \in E$ in addition to $p_{m,n}$, where $r_{m,n}$ is referred to as the *time-delay parameter* through link (m, n). The diffusion process unfolds in continuous-time t, and proceeds from a given initial active set S in the following way. Suppose that a node m becomes active at time t. Then, m is given a single chance to activate each currently inactive child node n. We choose a delay-time δ from the exponential distribution[1] with parameter $r_{m,n}$. If n has not been activated before time $t+\delta$, then m attempts to activate n, and succeeds with probability $p_{m,n}$. If m succeeds, then n will become active at time $t+\delta$. Said differently, whichever parent m that succeeds in satisfying the activation condition and for which the activation time is the earliest considering the time delay associated with each link can actually activate the node. Under the continuous time framework, it is unlikely that n is activated simultaneously by its multiple parent nodes exactly at time $t+\delta$. So we ignore this possibility. Whether or not m succeeds, it cannot make any further attempts to activate n in subsequent rounds. The process terminates if no more activations are possible.

4.1.2 Asynchronous Linear Threshold Model

Same as the above, we first recall the LT model. In this model, for every node $n \in V$, we specify a *weight* $(q_{m,n} > 0)$ from its parent node m in advance such that

$$\sum_{m \in B(n)} q_{m,n} \leq 1.$$

The diffusion process from a given initial active set S proceeds according to the following randomized rule. First, for any node $n \in V$, a *threshold* θ_n is chosen uniformly at random from the interval $[0, 1]$. At time-step t, an inactive node n is influenced by each of its active parent nodes, m, according to weight $q_{m,n}$. If the total weight from active parent nodes of n is no less than θ_n, that is,

$$\sum_{m \in B_t(n)} q_{m,n} \geq \theta_n,$$

[1] Similar formulation can be derived for other distributions such as power-law and Weibull.

then n will become active at time-step $t+1$. Here, $B_t(n)$ stands for the set of all the parent nodes of n that are active at time-step t. The process terminates if no more activations are possible.

The AsLT model is defined in a similar way to the AsIC. In the AsLT model, in addition to the weight set $\{q_{m,n}\}$, we specify real values $r_{m,n}$ with $r_{m,n} > 0$ in advance for each link (m, n). Same as for AsIC, we refer to $r_{m,n}$ as the *time-delay parameter* through link (m, n). The diffusion process unfolds in continuous-time t, and proceeds from a given initial active set S in the following way. Each active parent m of the node n exerts its effect on n with the time delay δ drawn from the exponential distribution with the delay parameter $r_{m,n}$. Suppose that the accumulated weight from the active parents of node n has become no less than θ_n at time t for the first time. Then, the node n becomes active at t without any delay and exerts its effect on its child with a delay associated with its link. This process is repeated until no more activations are possible.

4.2 Visualization Method

Let $R = \{(m, t_m), (n, t_n), \cdots\}$ be an information diffusion result over an undirected $G = (V, E)$, where (n, t_n) is a pair of an activated node and its activation time. We set the initial activation time to 0. From the set of nodes that appear in R, i.e., $V' = \{n \mid (n, t_n) \in R\}$, we obtain an induced subgraph $G' = (V', E')$. Here, we regard t_n as n's associated value for $n \in V'$. If $m \in V'$, $n \in V'$, $(m, n) \in E'$, and $t_m < t_n$, the direction of information diffusion is limited to from node m to n. Namely, a directed acyclic graph (DAG) is constructed from the information diffusion result R. Although our embedding method is designed for undirected graph, we can interpret that the diffusion of information takes over from the origin to the periphery by setting the radius of node n to t_n. The major reason why we restricted the graph we handle to undirected graph is to maintain clear meaning of the objective function we are trying to maximize. Alternatively, we can start with a directed graph and obtain a directed induced subgraph. Then we reinterpret it as an undirected subgraph, and apply the above discussion.

5. EXPERIMENTAL EVALUATION

5.1 Datasets

We generated diffusion results using both the AsIC and the AsLT models for four large real social networks. They are all bidirectionally connected networks. The first one is a trackback network of Japanese blogs used in [7]. It has 12,047 nodes and 79,920 directed links (the blog network). The second one is a coauthorship network used in [12], which has 12,357 nodes and 38,896 directed links (the coauthorship network). The third one is a network derived from the Enron Email Dataset [9] by extracting the senders and the recipients and linking those that had bidirectional communications. It has 4,254 nodes and 44,314 directed links (the Enron network). The fourth one is a network of people that was derived from the "list of people" within Japanese Wikipedia, used in [6], and has 9,481 nodes and 245,044 directed links (the Wikipedia network).

5.2 Experimental Results

We visualized the information diffusion result in 2-dimensional Euclidean space, i.e., $K = 2$, and compared the results of the proposed method with the other four existing methods. The initial active node was chosen to be the most influential node for each diffusion model. The location of this node is the origin of the visualization plane for the proposed method, but the location of the same node for the other methods is not controllable and determined by the algorithm of each method. The proposed method has the time information. A family of blue dotted circles of different radii centered at the origin indicates the activation times, where the radius t of each blue dotted circle corresponds to the actual time t. For all the visualization methods, red points and green lines are used to display the activated nodes and their links, respectively. It is noted that we are visualizing from the observed data, meaning that we don't know the parent which activated its child if there is more than one active parent. Thus, all the links between the activate parents and their active children are displayed.

Due to the space limitation, we only show parts of the results. Figure 1 shows the visualization result of information diffusion for the AsIC model over the Blog network using the proposed method, where the thick black circle indicate the initial active node. It is clear that the proposed method have the following properties:

1. Given two active nodes, we can easily see which one became active earlier.

2. Given an active node, we can easily identify its parents that could activate it (but we cannot identify it if there are multiple active parents by the reason mentioned above).

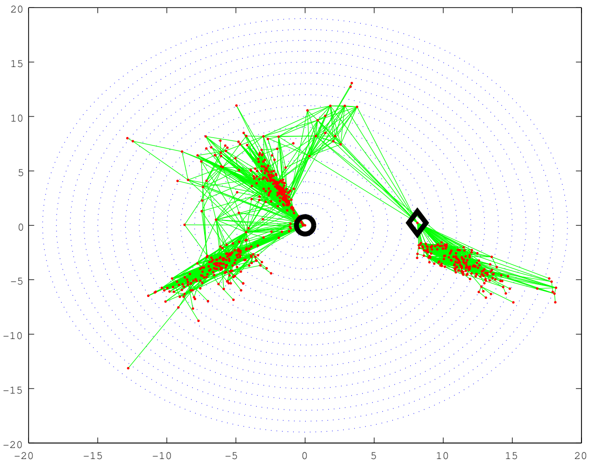


Figure 1: Visualization result of proposed method for the Blog network (AsIC model).

We can observe that in general *super-mediators*, i.e., those nodes that play an important role in passing the information to other nodes, are easily identified by the proposed method. In Figure 1, the thick black diamond node can naturally be interpreted as a super-mediator. The same node is also displayed as thick black diamonds in Figures 2, 3, 4 and 5. We notice that the multi-dimensional scaling and the spring force embedding methods are also good to find super-mediators, while it is more difficult to find them for the spectral embedding and the cross-entropy embedding methods. Note that the multi-dimensional scaling and the spring force

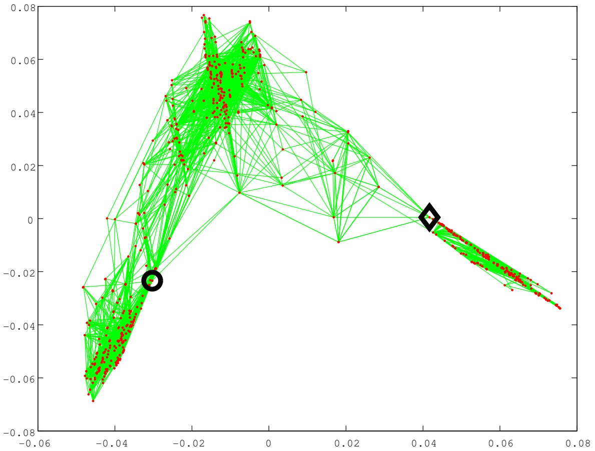


Figure 2: Visualization result of multi-dimensional scaling for the Blog network (AsIC model).

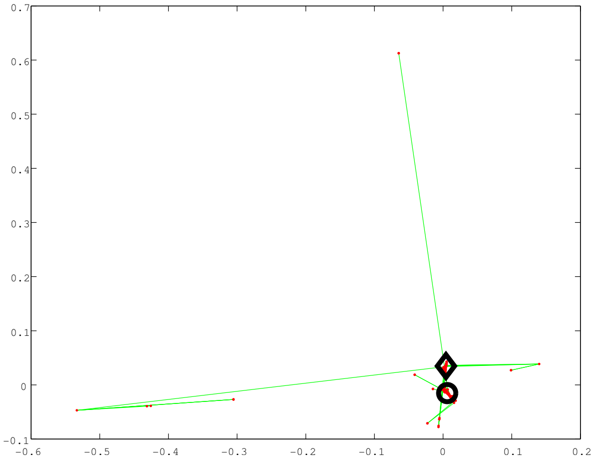


Figure 3: Visualization result of spectral embedding for the Blog network (AsIC model).

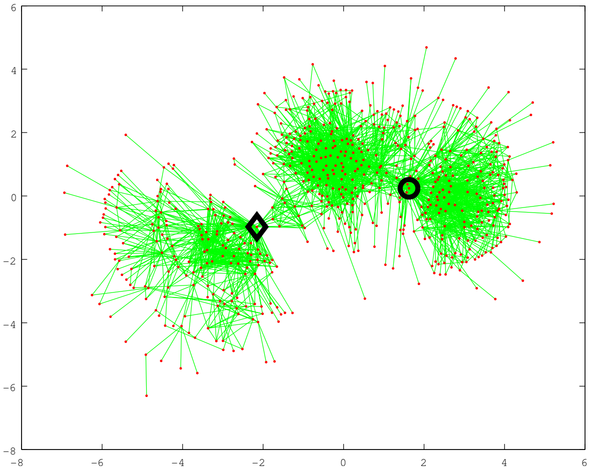


Figure 4: Visualization result of spring force embedding for the Blog network (AsIC model).

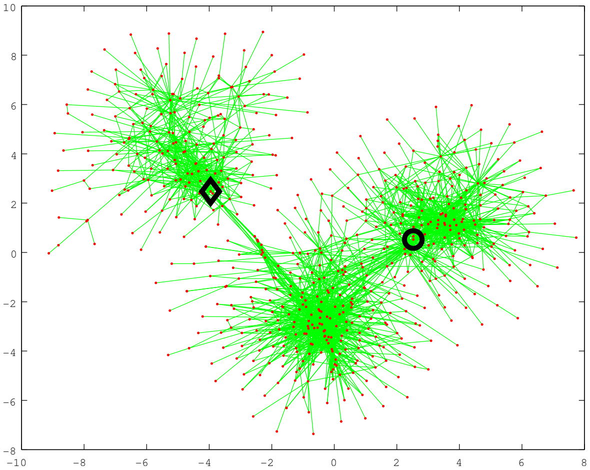


Figure 5: Visualization result of cross-entropy embedding for the Blog network (AsIC model).

embedding methods are based on graph distance matrix **G**, and the spectral embedding and the cross-entropy embedding methods are based on graph adjacency matrix **A**. For the **G**-based methods, the distance from the initial active node (thick black circle) to an active node v in the visualization plane can be correlated with the time if the node v is an active node. Thus, we can consider that such methods have a possibility of finding super-mediators. However, we see from Figures 1 to 5 that the proposed method better identifies a super-mediator than the multi-dimensional scaling and the spring force embedding methods.

Figure 6 shows the visualization result of information diffusion for the AsLT model over the Blog network. Compared with the visualization result for the AsIC model, we observe that links are mostly outward directed and only small links are in circumferential direction. We consider that this observation comes from a characteristic difference between the AsIC and AsLT models. Especially, in case of the AsLT model, when a parent node becomes active, only its low degree child nodes are likely to be activated. Our proposed method will locate these child nodes to similar directions because their connectivity patterns are necessarily close. We consider that this fact partly explains the difference between the visualization results of Figure 1 and 6.

Figures 7 and 8 respectively show the visualization results of information diffusion for the AsIC and AsLT models over the Wikipedia network. We can also see from these figures that the proposed method is promising for identifying influential super-mediators and exploring the characteristic differences between the two information diffusion models. As mentioned earlier, the visualization results over the coauthorship and Enron networks are omitted due to the space limitation, but it is confirmed that we obtained similar results.

Last but not least, we evaluated our proposed method only in the case of two-dimensional embedding for our visualization purpose, but this does not mean that it is limited to two-dimensional embedding. It is quite easy to extend it to the general K-dimension embedding. We plan to evaluate our method as a powerful technique for both dimensional reduction and clustering as a future work.

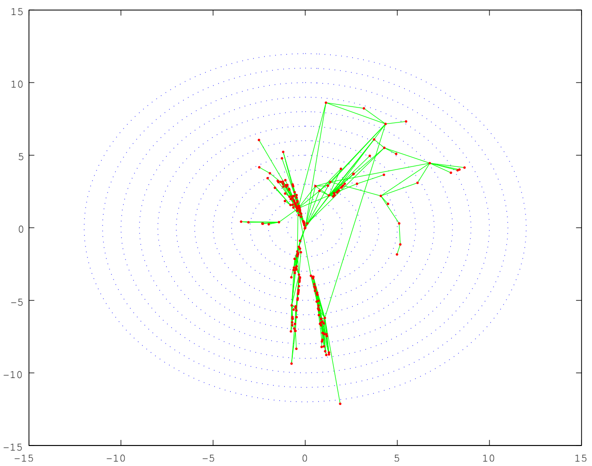


Figure 6: Visualization result of proposed method for the Blog network (AsLT model).

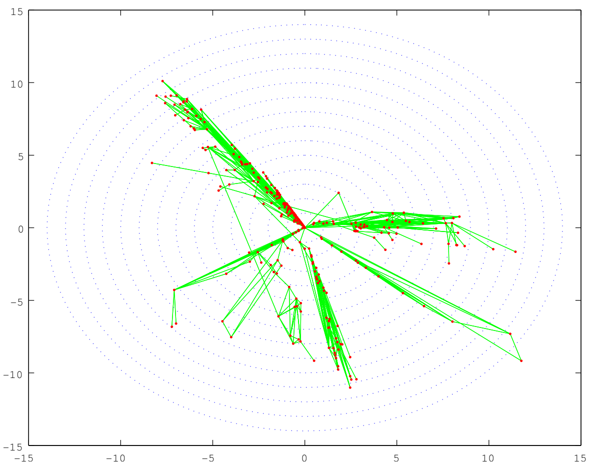


Figure 8: Visualization result of proposed method for the Wikipedia network (AsLT model).

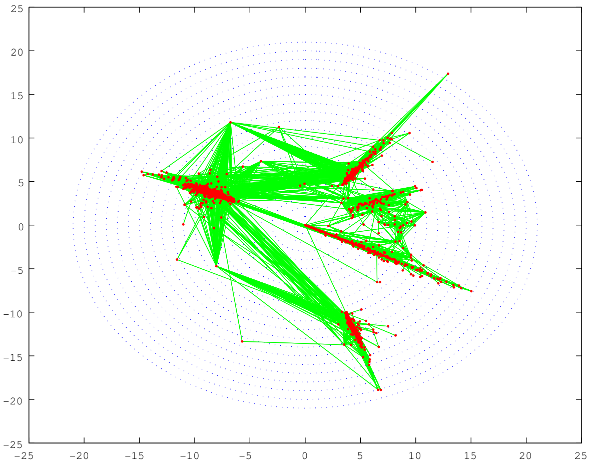


Figure 7: Visualization result of proposed method for the Wikipedia network (AsIC model).

6. DISCUSSION

One of the unique features of the proposed method is that we deal with the graph that has a value to each node, and the visualization takes account of the node value. The application to information diffusion involves time evolution and assigning the time the node gets activated to the node value works nicely to allow the diffusion starts at the origin always. On the contrary, all the other existing methods, when applied to the same visualization problem, generates a graph where the starting point of the diffusion is determined by the algorithm. Thus, if we want to visualize multiple results of diffusion sequences each starting from the same node, the starting node in each visualization is placed in a different location. Thus, the above feature is one of the advantages of the proposed method.

7. CONCLUSION

We addressed the problem of visualizing structure of a undirected graph that has a value associated with each node into a K-dimensional Euclidean space in such a way that 1) the length of the point vector in this space is equal to the value assigned to the node and 2) nodes that are connected are placed as close as possible to each other in the space and nodes not connected are placed as far apart as possible from each other. We showed that this visualization problem is reduced to spherical embedding that is formulated as a non-linear optimization problem for which a certain objective function to be maximized is defined. We proposed a very efficient algorithm based on a power iteration that employs double-centering operations. To validate the effectiveness of the proposed method, we applied it to visualize the information diffusion process over a social network by assigning the node activation time to the node value. We used the result of information diffusion obtained by two different diffusion models (AsIC and AsLT models) for four real world networks, and compared the proposed method with the multi-dimensional scaling, the spring force embedding, the spectral embedding and the cross-entropy embedding methods. We first confirmed that the proposed method can visualize time evolution of the diffusion process in an more intuitively understandable manner. We also confirmed that the proposed method have the following properties: 1) given two active nodes, we can easily see which one became active earlier, and 2) given an active node, we can easily identify its parents that could activate it (note that we are visualizing from the observed diffusion data, meaning that we don't know the parent which activated its child if there is more than one active parent.) Furthermore, we experimentally showed that the proposed method can better identify *supermediators*, i.e., those nodes that play an important role in passing the information to other nodes, than the other four methods.

8. ACKNOWLEDGMENTS

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-11-4111, JSPS Grant-in-Aid for Scientific Research (C) (No. 23500312), and Toyota Central R&D Labs., Inc. (No. E11114).

9. REFERENCES

[1] S. Brin and L.Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998.

[2] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, New York, 1997.

[3] T. Fushimi, Y. Kubota, K. Saito, M. Kimura, K. Ohara, and H. Motoda. Speeding up bipartite graph visualization method. In *Proceedings of the 24th Australasian Joint Conference on Artificial Intelligence*, pages 697–706. LNAI 7106, 2011.

[4] K. Kamada and S. Kawai. An algorithm for drawing general undirected graph. *Information Processing Letters*, 31:7–15, 1989.

[5] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pages 137–146, 2003.

[6] M. Kimura, K. Saito, and H. Motoda. Minimizing the spread of contamination by blocking links in a network. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, pages 1175–1180, 2008.

[7] M. Kimura, K. Saito, and H. Motoda. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data*, 3:9:1–9:23, 2009.

[8] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46:604–632, 1999.

[9] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *Proceedings of the 2004 European Conference on Machine Learning (ECML'04)*, pages 217–226, 2004.

[10] A. N. Langville and C. D. Meyer. Deeper inside pagerank. *Internet Mathematics*, 1:335–380, 2005.

[11] A. Naud, S. Usui, N. Ueda, and T. Taniguchi. Visualization of documents and concepts in neuroinformatics with the 3d-se viewer. *Frontiers in Neuroinformatics*, 1:Article 7, 2007.

[12] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[13] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Learning continuous-time information diffusion model for social behavioral data analysis. In *Proceedings of the 1st Asian Conference on Machine Learning (ACML2009)*, pages 322–337. LNAI 5828, 2009.

[14] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Selecting information diffusion models over social networks for behavioral analysis. In *Proceedings of the 2010 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2010)*, pages 180–195. LNAI 6323, 2010.

[15] W. Torgerson. *Theory and methods of scaling*. Wiley, New York, 1958.

[16] T. Yamada, K. Saito, and N. Ueda. Cross-entropy directed embedding of network data. In *Proceedings of the 20th International Conference on Machine Learning (ICML2003)*, pages 832–839, 2003.

Effect of In/Out-Degree Correlation on Influence Degree of Two Contrasting Information Diffusion Models

Kouzou Ohara[1], Kazumi Saito[2], Masahiro Kimura[3], and Hiroshi Motoda[4]

[1] Department of Integrated Information Technology, Aoyama Gakuin University
ohara@it.aoyama.ac.jp
[2] School of Administration and Informatics, University of Shizuoka
k-saito@u-shizuoka-ken.ac.jp
[3] Department of Electronics and Informatics, Ryukoku University
kimura@rins.ryukoku.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
motoda@ar.sanken.osaka-u.ac.jp

Abstract. How the information diffuses over a large social network depends on both the model employed to simulate the diffusion and the network structure over which the information diffuses. We analyzed both theoretically and empirically how the two contrasting most fundamental diffusion models, Independent Cascade (IC) and Linear Threshold (LT) behave differently or similarly over different network structures. We devised two rewiring structures, one preserving in/out-degree correlation and the other changing in/out-degree correlation while both preserving their in/out-degree distributions, and analyzed how co-link rate and in/out-degree correlation affect the influence degree of each diffusion model using two real world networks, each as the base network on which rewiring is imposed. The results of the theoretical analysis qualitatively explain the empirical results, and the findings help deepen the understanding of complex diffusion phenomena.

Keywords: Information diffusion, network structure, influence degree, node degree distribution

1 Introduction

The emergence of Social Media such as Facebook, Digg and Twitter has provided us with the opportunity to create large social networks, which are becoming an important medium for spreading information. Recently, substantial attention has been devoted to analyzing and mining social networks from the point of information diffusion [14, 15, 11, 19, 2, 1, 16]. One of the most well studied problems is the influence maximization problem, *i.e.*, the problem of finding a limited number of influential nodes that are effective for the spread of information. Many algorithms have been proposed to solve the problem using probabilistic information diffusion models on a network [8, 12, 5, 9, 6, 4]. In order to investigate diffusion phenomena using probabilistic models, it is indispensable to understand the behavioral differences among models, and provide an effective method for selecting the most appropriate model for a particular task we want to analyze.

There are two contrasting fundamental probabilistic models that have been widely used by many researchers. One is the *independent cascade (IC)* model [7, 8] and the other is the *linear threshold (LT)* model [18, 8]. The IC model takes a sender-centered approach such that each information sender independently influences its neighbors with some probability (*information push style model*). The LT model is a receiver-centered approach such that each information receiver adopts the information if and only if the number of its neighbors that have adopted the information exceeds some threshold, where the threshold is treated as a random variable (*information pull style model*). We analyze how the IC and the LT models differ from or similar to each other in terms of information diffusion for a wide range of social networks with different structures.

In this paper, we compare *influence degree* obtained by the IC and the LT models from the network structure perspective. Here, the influence degree of a node v under a probabilistic diffusion model in a network is defined to be the expected number of *active* nodes at the end of the information diffusion process that starts from the initial active node v, where nodes that have been influenced with the information are referred to as being active. First, we theoretically analyze the properties of the IC and the LT models on scale-free networks, and derive the following two properties: 1) as the in/out-degree correlation decreases, the influence degree decreases for the IC model but it does not change for the LT model and 2) as the co-link (bidirectional link) rate decreases, the influence degree increases for both the IC and the LT models, but the IC model is much less sensitive than the LT model. To verify these properties, we systematically generated a series of scale-free networks with varying in/out-degree correlation and co-link rate, applying two rewiring strategies, one preserving in/out-degree correlation and the other changing in/out-degree correlation while both preserving their in/out-degree distributions. We used two real world scale free networks as the bases to apply these strategies, and experimentally confirmed that the above two properties indeed hold.

2 Diffusion Models

Let $G = (V, E)$ be a directed network, where V and E $(\subset V \times V)$ are the sets of all the nodes and links, respectively, and $|V| \leq |E|$ can be naturally assumed for commonly-seen social networks. We recall the definition of the IC and the LT models according to the literatures [8, 9]. In these models, the diffusion process proceeds from an initial active node in discrete time-step $t \geq 0$, and it is assumed that nodes can switch their states only from inactive to active (*i.e.*, the SIR setting).

The IC model has a *diffusion probability* $p_{u,v}$ with $0 < p_{u,v} < 1$ for each link (u, v) as a parameter. Suppose that a node u first becomes active at time-step t, it is given a single chance to activate each currently inactive child node v, and succeeds with probability $p_{u,v}$. If u succeeds, then v will become active at time-step $t + 1$. If multiple parent nodes of v first become active at time-step t, then their activation trials are sequenced in an arbitrary order, but all performed at time-step t. Whether u succeeds or not, it cannot make any further trials to activate v in subsequent rounds. The process terminates if no more activations are possible.

The LT model has a *weight* $q_{u,v}$ (> 0) with $\sum_{u \in B(v)} q_{u,v} \leq 1$ for each link (u, v) as a parameter, where $B(v) = \{u \in V;\ (u, v) \in E\}$ is the set of parent nodes of node v. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0, 1]$. An inactive node v is influenced by its active parent nodes. If the total weight from v's active parent nodes at time-step t is no less than θ_v, *i.e.*, $\sum_{u \in B_t(v)} q_{u,v} \geq \theta_v$, then v will get activated at time-step $t+1$. Here, $B_t(v)$ is the set of all the parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

3 Analysis of Local Influence Degree

We first define local influence degree of node u, denoted by $\sigma_L(u)$, as the expected number of u's child nodes directly activated by u. For the IC model, $\sigma_L^{IC}(u)$ is given by $\sigma_L^{IC}(u) = \sum_{v \in F(u)} p_{u,v}$, where $F(u)$ stands for the set of u's child nodes defined by $F(u) = \{v \in V;\ (u, v) \in E\}$. For the LT model $\sigma_L^{LT}(u)$ is given by $\sigma_L^{LT}(u) = \sum_{v \in F(u)} q_{u,v}$ because each weight $q_{u,v}$ is regarded to be the probability that the threshold θ_v is chosen from the interval $[0, q_{u,v}]$. Then, we can calculate the average local influence degree over all nodes, denoted by $\bar{\sigma}_L(G)$. For the LT model, if we impose the condition $\sum_{u \in B(v)} q_{u,v} = 1$ for any node $v \in V$, we can prove $\bar{\sigma}_L^{LT}(G) = 1$ from the following relations.

$$\bar{\sigma}_L^{LT}(G) = \frac{1}{|V|} \sum_{u \in V} \sigma_L^{LT}(u) = \frac{1}{|V|} \sum_{u \in V} \sum_{v \in F(u)} q_{u,v} = \frac{1}{|V|} \sum_{(u,v) \in E} q_{u,v} = \frac{1}{|V|} \sum_{v \in V} \sum_{u \in B(v)} q_{u,v} \ = \ 1.$$

For the IC model, if we impose the uniform diffusion probability setting, i.e., $p_{u,v} = p$ for any link $(u, v) \in E$, which has been employed in many previous studies (*e.g.*, [8]), we can calculate $\bar{\sigma}_L^{IC}(G)$ as follows:

$$\bar{\sigma}_L^{IC}(G) = \frac{1}{|V|} \sum_{u \in V} \sigma_L^{IC}(u) = \frac{1}{|V|} \sum_{u \in V} \sum_{v \in F(u)} p_{u,v} = \frac{1}{|V|} \sum_{(u,v) \in E} p \ = \ \frac{|E|}{|V|} p,$$

where $\frac{|E|}{|V|}$ is equal to the average degree $d = \frac{1}{|V|} \sum_{u \in V} |B(u)| = \frac{1}{|V|} \sum_{u \in V} |F(u)| = \frac{|E|}{|V|}$, and is no less than 1 as we assume $|V| \leq |E|$. Thus, by setting the uniform diffusion probability to the inverse of average degree, i.e., $p = \frac{1}{d} = \frac{|V|}{|E|}$, we obtain $\bar{\sigma}_L^{IC}(G) = 1$. This makes the IC and LT models equivalent in terms of the average local influence degree. Hereafter, we impose these settings to evaluate the influence degree. Note that local influence degree of node u for the IC model becomes $\sigma_L^{IC}(u) = \sum_{v \in F(u)} p_{u,v} = p|F(u)|$.

So far we focused on local influence degree of node $u \in V$ under the condition that the node u has become active. However, when considering the cascade of information diffusion, we need to consider the probability $r(u)$ that the node u is activated by its parent nodes. Namely, we consider cascading local influence degree defined by $\sigma_{CL}(u) = r(u)\sigma_L(u)$. As the simplest case, we employ the probability $r(u)$ that the node u is activated at the next time step by some active node selected uniformly at random from the node set V. For the IC model, $r^{IC}(u)$ is given by $r^{IC}(u) = \frac{1}{|V|} \sum_{s \in B(u)} p_{s,u} = \frac{p|B(u)|}{|V|}$, and for the LT model, $r^{LT}(u)$ is given by $r^{LT}(u) = \frac{1}{|V|} \sum_{s \in B(u)} q_{s,u} = \frac{1}{|V|}$. Thus we obtain the average cascading local influence degree $\bar{\sigma}_{CL}$ for the IC and LT models as follows:

$$\bar{\sigma}_{CL}^{IC}(G) = \frac{1}{|V|} \sum_{u \in V} r^{IC}(u)\sigma_L^{IC}(u) = \frac{p^2}{|V|^2} \sum_{u \in V} |B(u)||F(u)|, \tag{1}$$

$$\bar{\sigma}_{CL}^{LT}(G) = \frac{1}{|V|} \sum_{u \in V} r^{LT}(u)\sigma_L^{LT}(u) = \frac{1}{|V|^2} \sum_{u \in V} \sigma_L^{LT}(u) = \frac{1}{|V|}. \tag{2}$$

Therefore, by noting that the in/out-degree correlation $dc_{I/O}(G)$ is quantified by

$$dc_{I/O}(G) = \frac{\frac{1}{|V|}\sum_{u \in V} |B(u)||F(u)| - d^2}{\sqrt{\frac{1}{|V|}\sum_{u \in V} |B(u)|^2 - d^2}\sqrt{\frac{1}{|V|}\sum_{u \in V} |F(u)|^2 - d^2}},$$

and the denominator of $dc_{I/O}(G)$ is determined by the standard deviations of in/out-degree distributions, we can see that the average cascading local influence degree of the IC model is affected by the in/out-degree correlation $dc_{I/O}(G)$ when the standard deviations are fixed, as shown in Eq. (1), while that of the LT model is not affected, as shown in Eq. (2). Namely, we can conjecture that influence degree of the IC model also decreases when the in/out-degree correlation decreases.

Another important factor affecting influence degree is the co-link rate $cr(G)$ which is defined by $cr(G) = \frac{1}{|E|}\sum_{u \in V} |B(u) \cap F(u)|$. Evidently, for a bidirectional network G, we obtain $cr(G) = 1$ because $B(u) = F(u)$ for any $u \in V$. Assume a node $v \in B(u) \cap F(u)$; if v succeeds activating u, then the reverse link (u, v) never contributes to increasing an active node, conversely, if u succeeds activating v, then the reverse link (v, u) never does so. Thus, we conjecture that influence degree of the IC and LT model increases when the co-link rate $cr(G)$ decreases. However, there is a subtle difference between the IC and the LT models. Think of the network with co-link rate close to 1. Evidently the in/out-degree correlation is also close to 1. Assume that k parents of a node v which has a large degree $D = |F(v)| = |B(v)|$ get activated. The expected probability that the node v becomes activated is $1 - (1 - 1/d)^k$ for the IC model and k/D for the LT model where d is the average node degree. For the IC model the probability is large for a small number of k and insensitive to $|D|$. Thus, once it gets activated, the reverse k links which do not contribute further activation is small. On the other hand, for the LT model the node v is not activated unless k is large. Thus, once it gets activated, the reverse k links do not contribute further activation is also large. This implies that the IC model is less sensitive to the change of co-link rate than the LT model.

4 Experiments

To confirm our conjectures in Section 3, we conducted extensive experiments using both synthetic and real world large networks, rewiring their links according to the two strategies presented in this section. However, due to the page limitation, we show only the results for the two real world networks: one bidirectional and the other directional[1].

4.1 Rewiring Strategies

We devised two rewiring strategies. Both preserve the in/out-degree distribution. The first one rewires links of a given network G preserving the in/out-degrees of each node, which is equivalent to the method of generating randomized networks presented in [13]. We implemented this strategy by swapping the two destination nodes v and v' of links

[1] The networks we omitted here include synthetic networks generated by the BA model [3] and the CNN model [17], and four other networks derived from the real world data.

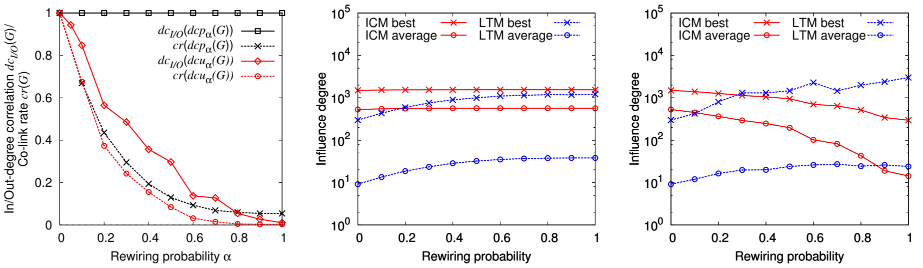
$e = (u, v)$ and $e' = (u', v')$ from two starting nodes u and u'. The links are chosen uniformly at random. Obviously, this never changes $dc_{I/O}(G)$, but does change $cr(G)$. We refer to this rewiring strategy as the DCP (in/out-Degree Correlation Preserved) method, and denote the network G rewired by this method by $dcp_\alpha(G)$, where α is the link rewiring probability, *i.e.*, v of e and v' of e' are swapped with the probability α. The larger α is, the smaller $cr(G)$ is. Thus, the DCP method allows us to investigate how the co-link rate affects the influence degree of the IC and the LT models. The second one rewires links changing the in/out-degree correlation. This is to confirm our conjecture that the in/out-degree correlation affects the influence degrees of the IC model. We implemented this by swapping $E_I(v)$, all the incoming links to a node v, and $E_I(v')$, all the incoming links to a node v' with a probability α. Nodes v and v' are randomly chosen. Namely, $E_I(v)$ becomes $\{(u, v); u \in B(v')\}$, and $E_I(v')$ becomes $\{(s, v'); s \in B(v)\}$ after swapping. This method changes the in-degree of chosen nodes without changing their out-degree while preserving the in/out-degree distributions of the network G. We refer to this method as the DCU (in/out-Degree Correlation Unpreserved) method, and denote the network G rewired by the DCU method with a link rewiring probability α by $dcu_\alpha(G)$. The larger α is, the smaller the in/out-degree correlation is.

4.2 Datasets and Network Structure

In this section, we explain the two real world networks for which we present the experimental results. The first one is a bidirectional network derived from the Enron Email Dataset [10]. We regarded each email address as a node, and constructed a bidirectional link between two email addresses u and v only if u sent an email to v and received an email from v. After that, we extracted the maximal strongly connected component. We refer to this bidirectional network as the Enron network, which has $4,254$ nodes and $44,314$ directed links. The second one is a directional network derived from a Japanese word-of-mouth communication site for cosmetics, "@cosme"[2], where each user page is associated with *fan links*. A fan link from user u to user v is generated if user v registers user u as his/her favorite user. We extracted a fan network from @cosme by tracing up to ten steps in the fan links starting from a randomly chosen user in December 2009. The resulting network has $45,024$ nodes and $351,299$ directed links. We refer to this network as the Cosme network.

For these networks, we investigated the influence degree $\sigma(v)$ of each node v of the networks $dcp_\alpha(G)$ and $dcu_\alpha(G)$ under the IC and the LT models, varying α from 0.0 to 1.0 by 0.1. Note that $dcp_{0.0}(G) = dcu_{0.0}(G) = G$. The influence degree $\sigma(v)$ was estimated by the empirical mean of the number of active nodes obtained from 10,000 independent runs of information diffusion based on the bond percolation technique [9]. According to the discussion in Section 3, we set a unique value $p = 1/d$ to every $p_{u,v}$ for the IC model. Namely, p was set to 0.10 for the Enron network, and 0.13 for the Cosme network.

[2] `http://www.cosme.net/`

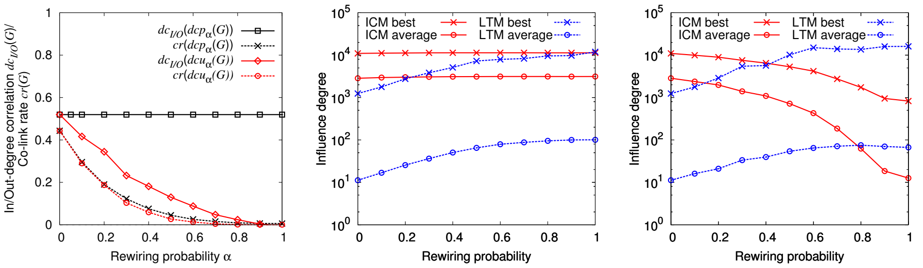


(a) The change of In/Out-degree correlation and Co-link rate.

(b) The best and the average influence degrees with the DCP method.

(c) The best and the average influence degrees with the DCU method.

Fig. 1: Experimental results for the Enron network.



(a) The change of In/Out-degree correlation and Co-link rate.

(b) The best and the average influence degrees with the DCP method.

(c) The best and the average influence degrees with the DCU method.

Fig. 2: Experimental results for the Cosme network.

4.3 Experimental Results

Figures 1a and 2a show how the in/out-degree correlation $dc_{I/O}(G)$ and the co-link rate $cr(G)$ of a given network G change with the two rewiring methods, DCP and DCU, for the Enron and the Cosme networks, respectively. We see that both methods work just as we intended: $cr(G)$ decreases in a similar fashion for both the DCP and the DCU methods, as the rewiring probability α becomes larger, while $dc_{I/O}(G)$ does not change with the DCP method, but it does decrease similarly to $cr(G)$ with the DCU method. Note that both $dc_{I/O}(G)$ and $cr(G)$ of the Enron network are 1.0 for $\alpha = 0.0$ because it is bidirectional.

Figure 1b illustrates how the DCP method affects the best and the average influence degrees over all the nodes of the Enron network. As we expected, both influence degrees of the LT model become larger as the rewiring probability becomes larger, and the co-link rate becomes smaller. The influence degrees of the IC model does not seem to increase, but indeed they slightly increase within the range of $\alpha = 0.0$ to 0.6 where the co-link rate drastically decreased. This qualitatively supports the analysis in Section 3.

The same tendencies can be found in the result for the Cosme network as shown in Fig. 2b. We also observed the same tendencies for the other networks we omitted here.

Figures. 1c and 2c show how the DCU method affects the best and the average influence degrees of the IC and the LT models. Both $dc_{I/O}(G)$ and $cr(G)$ decrease with α. This imposes two conflicting factors for the IC model, but the effect of $dc_{I/O}(G)$ surpasses and the influence degrees of the IC model decrease for both the Enron and the Cosme networks. On the other hand, the influence degrees of the LT model are affected by only $cr(G)$. Thus, they increase in the same way as in Figs. 1b and 2b. The same observation is obtained for the other networks. This also qualitatively supports the analysis in Section 3.

5 Conclusion

Understanding how information diffuses over a large social network is important to do any kind of social network analysis, but it is difficult because actual diffusion depends on both the diffusion model employed and the properties of the network structure over which the information diffuses. Independent Cascade (IC) and Linear Threshold (LT) models have been used widely by many researchers. Both are probabilistic models but have contrasting properties, *i.e.*, information push (IC) and information pull (LT). Social networks have common characteristics. The most important one would be the scale free property. There can be many structures that hold this property. We devised two rewiring strategies that can systematically transform one network structure to another structure preserving the scale free property, one preserving in/out-degree correlation (DCP method) and the other changing in/out-degree correlation (DCU method). Each strategy was successively applied with different probabilities to two real world social networks, generating a series of networks, each with a gradually changing structure. We chose co-link rate and in/out-degree correlation as the two parameters that characterize the network structure, and investigated how these parameters affects the influence degree of the two models (IC and LT). The major new findings are 1) the IC model is sensitive to in/out-degree correlation and the influence degree is positively correlated to it, whereas the LT model is insensitive to it and 2) Both the IC and the LT models are negatively correlated to co-link rate, but its dependency is much less sensitive in the IC model. These properties can be qualitatively derived by the theoretical analysis and verified by the extensive experiments using the above networks as well as others not reported in this paper. These findings are useful in deepening our understanding of the complex information diffusion phenomena over a social network.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-11-4111, and JSPS Grant-in-Aid for Scientific Research (No. 23700181).

References

1. Bakshy, E., Hofman, J., Mason, W., Watts, D.: Everyone's an influencer: Quantifying influences on twitter. In: Proceedings of the 4th International Conference on Web Search and Data Mining (WSDM2011). pp. 65–74 (2011)
2. Bakshy, E., Karrer, B., Adamic, L.A.: Social influence and the diffusion of user-created content. In: Proceedings of the 10th ACM conference on Electronic Commerce. pp. 325–334 (2009)
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
4. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010). pp. 1029–1038 (2010)
5. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009). pp. 199–208 (2009)
6. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of the 10th IEEE International Conference on Data Mining (ICDM 2010). pp. 88–97 (2010)
7. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters 12, 211–223 (2001)
8. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003). pp. 137–146 (2003)
9. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Mining and Knowledge Discovery 20, 70–97 (2010)
10. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Proceedings of the 2004 European Conference on Machine Learning (ECML'04). pp. 217–226 (2004)
11. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06). pp. 228–237 (2006)
12. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007). pp. 420–429 (2007)
13. Melo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: Simple building blocks of complex networks. Science 298, 824–827 (2002)
14. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E 66, 035101 (2002)
15. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002). pp. 61–70 (2002)
16. Romero, D., Meeder, B., Kleinberg, J.: Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In: Proceedings of the 20th International World Wide Web Conference (WWW2011). pp. 695–704 (2011)
17. Vázquez, A.: Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. Physical Review 67(5), 056104 (2003)
18. Watts, D.J.: A simple model of global cascades on random networks. Proceedings of National Academy of Science, USA 99, 5766–5771 (2002)
19. Watts, D.J., Dodds, P.S.: Influence, networks, and public opinion formation. Journal of Consumer Research 34, 441–458 (2007)

Speeding up Bipartite Graph Visualization Method

Takayasu Fushimi[1], Yamato Kubota[2], Kazumi Saito[1,2], Masahiro Kimura[3], Kouzou Ohara[4], and Hiroshi Motoda[5]

[1] Graduate School of Management and Information of Innovation, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
{j11507,k-saito}@u-shizuoka-ken.ac.jp
[2] School of Management and Information, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
{b08038, k-saito}@u-shizuoka-ken.ac.jp
[3] Department of Electronics and Informatics, Ryukoku University
Otsu, Shiga 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[4] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp
[5] Institute of Scientific and Industrial Research, Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We address the problem of visualizing structure of bipartite graphs such as relations between pairs of objects and their multi-labeled categories. For this task, the existing spherical embedding method, as well as the other standard graph embedding methods, can be used. However, these existing methods either produce poor visualization results or require extremely large computation time to obtain the final results. In order to overcome these shortcomings, we propose a new spherical embedding method based on a power iteration, which additionally performs two operations on the position vectors: double-centering and normalizing operations. Moreover, we theoretically prove that the proposed method always converges. In our experiments using bipartite graphs constructed from the Japanese sites of Yahoo!Movies and Yahoo!Answers, we show that the proposed method works much faster than these existing methods and still the visualization results are comparable to the best available so far.

1 Introduction

Visualization by embedding graphs into a low dimensional Euclidean space plays an important role to intuitively understand the essential structure of graphs (networks). To this end, various graph embedding methods have been proposed in the past that include multi-dimensional scaling [5], spectral embedding [1], spring force embedding [2], cross-entropy embedding [7]. Each method has its own advantages and disadvantages.

In this paper, we address the problem of visualizing structure of bipartite graphs such as relations between pairs of objects and their multi-labeled categories. More specifically, relations of this kind include pairs of movies and their associated genres, pairs of persons and their interested genres, pairs of researchers and their coauthoring papers, pairs of words and their appearing documents, and many more. Clearly, we can straightforwardly apply any one of the above-mentioned embedding methods for the visualization. However, we note that these standard methods have an intrinsic limitation because they cannot make much use of the essential structure of bipartite graphs. Indeed, the existing spherical embedding method has been proposed for the purpose of visualizing bipartite graphs [6]. In this method, the position vectors are embedded on two concentric spheres (circles) with different radii. We consider that such a spherical embedding can be a natural representation for bipartite graphs. However, the biggest problem with the existing method is that it often requires an extremely large computation time to obtain the final visualization results.

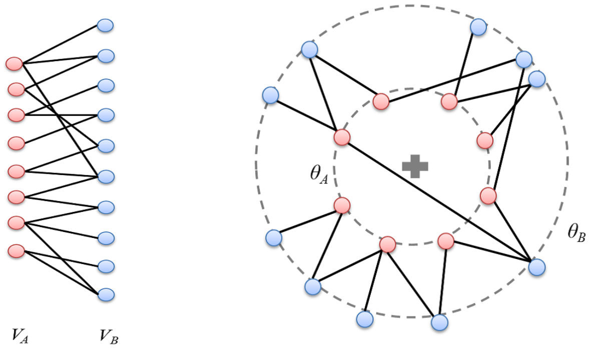
In this paper, to overcome these shortcomings, we propose a new spherical embedding method based on a power iteration, which adopts two operations to iteratively adjust the positioning vectors: double-centering and normalizing operations. We further show theoretically that the convergence of the proposed algorithm is always guaranteed. In our experiments that use bipartite graphs constructed from the Japanese sites of Yahoo!Movies and Yahoo!Answers, we show that the proposed method works much faster than these existing methods, and yet the visualization results are comparable to the best available so far.

This paper is organized as follows. We first describe the problem framework of embedding bipartite graphs into a low dimensional Euclidean space in Section 2. Next we describe our proposed algorithm, and prove that this algorithm always converges in Section 3. Then we experimentally evaluate the proposed method by comparing it with the existing embedding methods in terms of both the efficiency of the algorithms and ease of the interpretability of visualization results in Section 4. We last summarize the main conclusion in Section 5.

2 Problem Framework

We describe the problem framework of embedding the bipartite graph $G = (V, E)$ into a K-dimensional Euclidean space, where $V = V_A \cup V_B$, $V_A \cap V_B = \emptyset$, and $E \subset V_A \times V_B$. For the sake of technical convenience, we identify each set of the nodes, V_A and V_B, by two different series of positive integers, i.e., $V_A = \{1, \cdots, m, \cdots, M\}$ and $V_B = \{1, \cdots, n, \cdots, N\}$. Here M and N are the numbers of the nodes in V_A and V_B , i.e., $|V_A| = M$ and $|V_B| = N$, respectively. Then, we can define the $M \times N$ adjacency matrix $\mathbf{A} = \{a_{m,n}\}$ by setting $a_{m,n} = 1$ if $(m, n) \in E$; $a_{m,n} = 0$ otherwise. We denote the K-dimensional embedding position vectors by $\mathbf{x}_m$ for the node $m \in V_A$ and $\mathbf{y}_n$ for the node $n \in V_B$. Then we can construct $M \times K$ and $N \times K$ matrices consisting of these position vectors, i.e., $\mathbf{X} = (\mathbf{x}_1, \cdots \mathbf{x}_M)^T$ and $\mathbf{Y} = (\mathbf{y}_1, \cdots \mathbf{y}_N)^T$. Here $\mathbf{X}^T$ stands for the transposition of $\mathbf{X}$.

According to the work on the existing spherical embedding method [6], we explain the framework of spherical embedding of bipartite graph. In Fig. 1, we show an example in a two-dimensional Euclidean space, i.e., unlike the standard visualization scheme shown in Fig. 1a, we consider locating the position vectors on two concentric spheres (circles) as shown in Fig. 1b. We believe that this kind of spherical embedding is natural



(a) Bipartite Graph (b) Spherical Embedding

Fig. 1: Spherical Embedding for Bipartite Graph

to represent bipartite graphs, and its usefulness has been reported [6]. Hereafter, we assume that nodes in subset V_A are located on the inner circle θ_A with radius $r_A = 1$, while nodes in V_B are located on the outer circle θ_B with radius $r_B = 2$. Note that $\|\mathbf{x}_m\| = 1$, $\|\mathbf{y}_n\| = 2$. Then, our aim is to locate the position vectors of the nodes having similar connection patterns closely to each other.

3 Proposed Method

3.1 Proposed Algorithm

The new spherical embedding method is based on a power iteration. It has two operations on the positioning vectors which we call double-centering operation and normalizing operation. In order to describe our algorithm, we need to introduce the centering matrices and normalizing operations. The centering (Young-Householder transformation) matrices are defined as $\mathbf{H}_M = \mathbf{I}_M - \frac{1}{M}\mathbf{1}_M\mathbf{1}_M^T$, $\mathbf{H}_N = \mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^T$ where $\mathbf{I}_M$ and $\mathbf{I}_N$ stands for $M \times M$ and $N \times N$ identity matrices, respectively, and $\mathbf{1}_M$ and $\mathbf{1}_N$ are M- and N-dimensional vectors whose elements are all one. Clearly, the mean vector of the resulting position vectors becomes $\mathbf{0}$ by the operations $\mathbf{H}_M\mathbf{X}$ and $\mathbf{H}_N\mathbf{Y}$. On the other hand, the normalizing operations are defined as $\mathbf{\Lambda}_M(\mathbf{X}) = r_A\mathrm{diag}(\mathbf{X}\mathbf{X}^T)^{-1/2}\mathbf{X}$, $\mathbf{\Lambda}_N(\mathbf{Y}) = r_B\mathrm{diag}(\mathbf{Y}\mathbf{Y}^T)^{-1/2}\mathbf{Y}$, where diag$(\cdot)$ is an operation to set all the non-diagonal elements to zero, i.e., diag$(\mathbf{X}\mathbf{X}^T)$ is a diagonal matrix whose m-th element is $\mathbf{x}_m^T\mathbf{x}_m$.

Intuitively, the basic procedure of our proposed algorithm is that the position vector $\mathbf{x}_m$ is repeatedly moved to the position calculated by adding the position vectors $\{\mathbf{y}_n\}$ that are connected to $\mathbf{x}_m$. Of course, we need to perform a normalizing operation so as to satisfy the spherical constraints. Below we describe our proposed algorithm.

1. Initialize the matrix $\mathbf{X}$ and $\mathbf{Y}$.
2. Update the matrix $\mathbf{X} \leftarrow \mathbf{\Lambda}_M(\mathbf{H}_M\mathbf{A}\mathbf{H}_N\mathbf{Y})$.
3. Update the matrix $\mathbf{Y} \leftarrow \mathbf{\Lambda}_N(\mathbf{H}_N\mathbf{A}^T\mathbf{H}_M\mathbf{X})$.
4. Terminate if the changes for the position vectors $\mathbf{X}$ and $\mathbf{Y}$ are small.
5. Return to the step 2.

As the basic framework, our proposed algorithm employs a power iteration, just like the HITS algorithm [3], which utilizes $\mathbf{A}$ and $\mathbf{A}^T$, does. However, the main differences are use of the double-centering operations by $\mathbf{H}_M$ and $\mathbf{H}_N$ and the normalizing operations by $\mathbf{\Lambda}_M(\cdot)$ and $\mathbf{\Lambda}_N(\cdot)$. Here note that the double-centering operation is also employed in the standard multidimensional scaling method [5].

Now we briefly mention the computational complexity of our algorithm. Clearly, the main computational complexity of one-iteration comes from the multiplication by the matrix $\mathbf{A}$ (or $\mathbf{A}^T$) which is the most intensive part and is proportional to the number of links in the bipartite graph. Thus, the proposed algorithm is expected to work much faster especially for a sparse bipartite graph, compared with the existing spherical embedding algorithm that require a nonlinear optimization just like a spring force embedding [2] does. In fact, it has been well known that the PageRank algorithm based on a power iteration works very fast for a large and sparse network [4].

3.2 Convergence Proof

We prove the convergence property of the algorithm. To do this, we first introduce the following double-centered matrix $\mathbf{B} = \{b_{m,n}\}$ that is calculated from the adjacency matrix $\mathbf{A}$ as $\mathbf{B} = \mathbf{H}_M\mathbf{A}\mathbf{H}_N$. Then, by using the matrix $\mathbf{B}$, we can consider the following objective function with respect to the position vectors $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_M)^T$ and $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_N)^T$.

$$J(\mathbf{X}, \mathbf{Y}) = \sum_{m=1}^{M}\sum_{n=1}^{N} b_{m,n}\frac{\mathbf{x}_m^T}{r_A}\frac{\mathbf{y}_n}{r_B} + \frac{1}{2}\sum_{m=1}^{M}\lambda_m(r_A^2 - \mathbf{x}_m^T\mathbf{x}_m) + \frac{1}{2}\sum_{n=1}^{N}\mu_n(r_B^2 - \mathbf{y}_n^T\mathbf{y}_n), \quad (1)$$

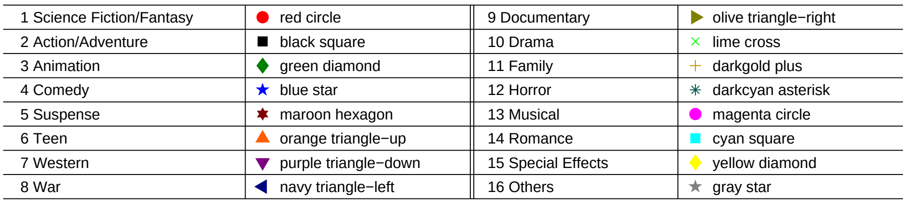
where $\{\lambda_m \mid m = 1, \cdots, M\}$ and $\{\mu_n \mid n = 1, \cdots, N\}$ correspond to Lagrange multipliers for the spherical constraints, i.e., $\mathbf{x}_m^T\mathbf{x}_m = r_A^2$ and $\mathbf{y}_n^T\mathbf{y}_n = r_B^2$ for $1 \leq m \leq M$ and $1 \leq n \leq N$.

Now we consider maximizing $J(\mathbf{X}, \mathbf{Y})$ defined in Equation (1) by use of a coordinate strategy. Note that maximizing $J(\mathbf{X}, \mathbf{Y})$ pushes the pairs $\mathbf{x}_m$ and $\mathbf{y}_m$ to the same direction if they are connected and pushes them to the opposite direction if they are unconnected, and realizes the intended visualization. We repeat the following two steps: maximizing $J(\mathbf{X}, \mathbf{Y})$ with respect to $\mathbf{X}$ by fixing the matrix $\mathbf{Y}$ first, and maximizing $J(\mathbf{X}, \mathbf{Y})$ with respect to $\mathbf{Y}$ by fixing the matrix $\mathbf{X}$ next. If the maximization of these steps are achieved by the above algorithm's step 2 and 3, respectively, we can guarantee the convergence of our proposed algorithm.

In order to confirm these facts, we consider the following gradient vector of the objective function $J(\mathbf{X}, \mathbf{Y})$ with respect to $\mathbf{x}_m$.

$$\frac{\partial J(\mathbf{X}, \mathbf{Y})}{\partial \mathbf{x}_m} = \frac{1}{r_A r_B}\sum_{n=1}^{N} b_{m,n}\mathbf{y}_n - \lambda_m\mathbf{x}_m. \quad (2)$$

Thus, for a fixed matrix $\mathbf{Y}$, we obtain the optimal position vector $\mathbf{x}_m$ which maximizes the objective function $J(\mathbf{X}, \mathbf{Y})$ as $\mathbf{x}_m = \frac{r_A}{\|\tilde{\mathbf{x}}_m\|}\tilde{\mathbf{x}}_m$, $\tilde{\mathbf{x}}_m = \sum_{n=1}^{N} b_{m,n}\mathbf{y}_n$. Here note that the optimal vector $\mathbf{x}_m$ is calculated by using the matrix $\mathbf{Y}$ only. Thus, for $m = 1, \cdots, M$, by using the normalizing operation $\mathbf{\Lambda}_M(\cdot)$ whose diagonal elements become



1 Science Fiction/Fantasy	● red circle	9 Documentary	▶ olive triangle−right
2 Action/Adventure	■ black square	10 Drama	× lime cross
3 Animation	◆ green diamond	11 Family	+ darkgold plus
4 Comedy	★ blue star	12 Horror	✳ darkcyan asterisk
5 Suspense	✶ maroon hexagon	13 Musical	● magenta circle
6 Teen	▲ orange triangle−up	14 Romance	■ cyan square
7 Western	▼ purple triangle−down	15 Special Effects	◆ yellow diamond
8 War	◀ navy triangle−left	16 Others	★ gray star

Fig. 2: category names in Japanese Yahoo!Movies site

$r_A/||\tilde{\mathbf{x}}_1||, \cdots, r_A/||\tilde{\mathbf{x}}_M||$, we can obtain the following solution in the vector-matrix representation.

$$\mathbf{X} = \mathbf{\Lambda}_M(\mathbf{BY}) = \mathbf{\Lambda}_M(\mathbf{H}_M\mathbf{A}\mathbf{H}_N\mathbf{Y}). \tag{3}$$

Recall that Equation (3) performs centering the matrix $\mathbf{Y}$ by the matrix $\mathbf{H}_N$, multiplies the adjacency matrix $\mathbf{A}$, performs re-centering the matrix by multiplying the matrix $\mathbf{H}_M$, and normalizes so as to guarantee spherical constraints (with radius r_A). By this formula, we can obtain the optimal solution of position vectors $\mathbf{X}$ by fixing the matrix $\mathbf{Y}$.

Similarly, we can also obtain the following optimal solution of position vector $\mathbf{y}_n$ by fixing the matrix $\mathbf{X}$ as $\mathbf{y}_n = \frac{r_B}{||\tilde{\mathbf{y}}_n||}\tilde{\mathbf{y}}_n$, $\tilde{\mathbf{y}}_n = \sum_{m=1}^{M} b_{m,n}\mathbf{x}_m$. Thus, for $n = 1, \cdots, N$, by using the normalizing operation $\mathbf{\Lambda}_N(\cdot)$ whose diagonal elements become $r_B/||\tilde{\mathbf{y}}_1||, \cdots, r_B/||\tilde{\mathbf{y}}_N||$, we can obtain the following solution in the vector-matrix representation.

$$\mathbf{Y} = \mathbf{\Lambda}_N(\mathbf{B}^T\mathbf{X}) = \mathbf{\Lambda}_N(\mathbf{H}_N\mathbf{A}^T\mathbf{H}_M\mathbf{X}). \tag{4}$$

Therefore, since the finite objective function $J(\mathbf{X}, \mathbf{Y})$ defined in Equation (1) has the analytical optimal solution under the condition that either $\mathbf{X}$ or $\mathbf{Y}$ is fixed, and is always maximized by performing the step 2 and 3 of the algorithm, we can guarantee that the algorithm always converges.

4 Evaluation by Experiments

4.1 Network Data

We regard the movies as nodes in V_B, and their genres as nodes in V_A for the Japanese Yahoo!Movies site [2]. Note that each movie is associated with more than or equal to one genre. In Fig. 2, we show their genre names used in our experiments, and for our visual analyses purpose, we assign an individual marker with a different color to each genre as shown in this figure. In order to evaluate our proposed method by using a set of different bipartite graphs, we classify these movies into 7 groups according to their release dates(1950-59, 1960-69, 1970-79, 1980-89, 1990-99, 2000-04 and 2005-09).Here the number of genres is $|V_A| = 16$ for all the periods, the number of movies $|V_B|$ are 594, 1079, 1314, 1805, 2659, 2948 and 3264, and the number of links $|E|$ are 899, 1617, 2071, 2994, 4424, 6057 and 6564 for each period.

[2] http://movies.yahoo.co.jp/

We regard the users who answered questions as nodes in V_B, and the genres of these questions as nodes in V_A for the Japanese Yahoo!Answers site [3]. Note that although each question belongs to only one genre, the same user frequently answers several questions belonging to a wide variety of genres. Thus we can obtain bipartite graphs between the pairs of the users and the genres they answered. In our experiments, we utilized a set of data from April, 2004, to October, 2005. Again, in order to evaluate our proposed method by using a set of different bipartite graphs, we classify these questions into 6 groups according to their submission dates().Here the number of genres is $|V_A| = 10$ for all the periods, the number of users $|V_B|$ are 11871, 27446, 35907, 39451, 42884 and 46834, and the number of links $|E|$ are 30849, 80664, 96926, 95714, 102086 and 112548 for each period.

4.2 Brief Description of Other Visualization Methods used for Comparison

We first explain the existing spherical embedding method as our primal comparison method, whose problem framework is the same to ours. In this method the following objective function is directly minimized with respect to the position vectors $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_M)^T$ and $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_N)^T$ under the constraints that $\mathbf{x}_m^T \mathbf{x}_m = r_A^2$ and $\mathbf{y}_n^T \mathbf{y}_n = r_B^2$ for $1 \leq m \leq M$ and $1 \leq n \leq N$. The objective function is defined as $\mathcal{J}(\mathbf{X}, \mathbf{Y}) = \frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{N} \left(c_{m,n} r_A r_B - \mathbf{x}_m^T \mathbf{y}_n \right)^2$, where $c_{m,n} = 2a_{m,n} - 1$, i.e., $c_{m,n} = 1$ if $(m, n) \in E$; $c_{m,n} = -1$ otherwise. In order to obtain the solution vectors, this method repeatedly moves each position vector by using the Newton method in a framework of nonlinear optimization, i.e., it repeats the following two steps: First, minimizing $\mathcal{J}(\mathbf{X}, \mathbf{Y})$ for $\mathbf{x}_m$ by fixing $\{\mathbf{x}_1, \cdots \mathbf{x}_M\} \setminus \mathbf{x}_m$ and $\{\mathbf{y}_1, \cdots \mathbf{y}_N\}$, and next minimizing $\mathcal{J}(\mathbf{X}, \mathbf{Y})$ for $\mathbf{y}_n$ by fixing $\{\mathbf{x}_1, \cdots \mathbf{x}_M\}$ and $\{\mathbf{y}_1, \cdots \mathbf{y}_N\} \setminus \mathbf{y}_n$. Thus this method requires an extremely large computation time to obtain the final results.

We have further compared the proposed method with the four well known embedding methods: multi-dimensional scaling [5], spectral embedding [1], spring force embedding [2], and cross-entropy embedding [7]. Here the former two perform a power iteration with respect to either a double-centered distance matrix or a graph Laplacian matrix which is calculated from a given graph, just like our proposed spherical embedding method does, while the latter two repeatedly move each position vector by using the Newton method in a framework of nonlinear optimization, just like the existing spherical embedding method does. Note that these four methods are not designed for embedding bipartite graphs, but as mentioned earlier, we can straightforwardly apply them for our purpose because a bipartite graph is regarded as an instance of general undirected graph.

In what follows in this subsection, we regard a bipartite graph as an undirected graph $G = (V, E)$ to describe the basic ideas of these standard embedding methods, and then consider a framework of embedding it into a K-dimensional Euclidean space. In this framework, we identify the set of the nodes by a positive integer, i.e., $V = \{1, \cdots, l, \cdots, L\}$, $|V| = L$ and $L = M + N$. Then, we can define the $L \times L$ adjacency matrix $\mathbf{A} = \{a_{m,n}\}$ by setting $a_{m,n} = 1$ if $(m, n) \in E$; $a_{m,n} = 0$ otherwise. We denote the K-dimensional embedding position vectors by $\mathbf{x}_m$ for the node $m \in V$, and then

[3] http://chiebukuro.yahoo.co.jp/

construct an $L \times K$ matrix consisting of these position vectors, i.e., $\mathbf{X} = (\mathbf{x}_1, \cdots \mathbf{x}_L)^T$. We also denote the graph distance matrix by $\mathbf{G} = \{g_{m,n}\}$, each element of which is the minimum path length between node m and node n.

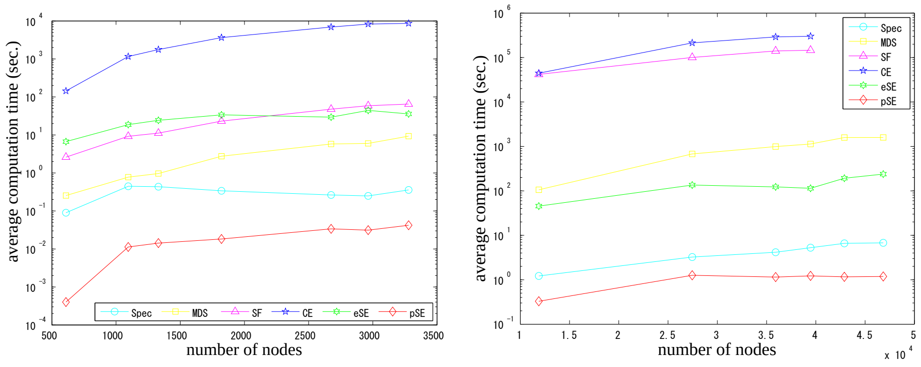
Multi-dimensional scaling method [5] first calculates the distance matrix $\mathbf{G}$, and performs the double centering operation ($\mathbf{H}_L = \mathbf{I}_L - \frac{1}{L}\mathbf{1}_L\mathbf{1}_L^T$)

to the distance matrix. Mathematically it is formulated as minimizing $\mathcal{M}(\mathbf{X}) = \frac{1}{2}\sum_{k=1}^{K} \mathbf{z}_k^T(\mathbf{H}_L\mathbf{G}\mathbf{H}_L)\mathbf{z}_k$, where $\mathbf{z}_k = (x_{1,k}, \cdots, x_{L,k})^T$, and $\{\mathbf{z}_1, \cdots, \mathbf{z}_K\}$ need to be orthonormal vectors, i.e., $\mathbf{z}_k^T\mathbf{z}_k = 1$ and $\mathbf{z}_k^T\mathbf{z}_{k'} = 0$ if $k \neq k'$. Spectral embedding method [1] tries to directly minimize distances between position vectors of connecting nodes. Mathematically it is formulated as minimizing $\mathcal{S}(\mathbf{X}) = \sum_{k=1}^{K} \mathbf{z}_k^T(\mathbf{D} - \mathbf{A})\mathbf{z}_k$, where $\mathbf{D}$ is a diagonal matrix each element of which is the degree of node (number of links). Note that $(\mathbf{D} - \mathbf{A})$ is referred to as a graph Laplacian matrix. Again, we set $\mathbf{z}_k = (x_{1,k}, \cdots, x_{L,k})^T$, and $\{\mathbf{z}_1, \cdots, \mathbf{z}_K\}$ need to be orthonormal vectors, which excludes the trivial vector expressed as $\mathbf{z} \propto \mathbf{1}_L$. Spring force embedding method [2] assumes that there is a hypothetical spring between each connected node pair and locates nodes such that the distance of each node pair is closest to its minimum path length at equilibrium. Mathematically it is formulated as minimizing $\mathcal{K}(\mathbf{X}) = \sum_{\mathbf{m=1}}^{\mathbf{L-1}} \sum_{\mathbf{n=m+1}}^{\mathbf{L}} \alpha_{\mathbf{m,n}}(\mathbf{g_{m,n}} - \|\mathbf{x_m} - \mathbf{x_n}\|)^{\mathbf{2}}$, where $\alpha_{m,n}$ is a spring constant which is normally set to $1/(2g_{u,v}^2)$. Cross-entropy embedding method [7] first defines a similarity $\rho(\mathbf{x}_m, \mathbf{x}_n)$ between the embedding positions x_m and x_n and uses the corresponding element $a_{m,n}$ of the adjacency matrix as a measure of distance between the node pair, and tries to minimize the total cross entropy between these two. Mathematically it is formulated as minimizing $C(\mathbf{X}) = -\sum_{\mathbf{m=1}}^{\mathbf{M-1}} \sum_{\mathbf{n=m+1}}^{\mathbf{M}} (\mathbf{a_{m,n}} \log \rho(\mathbf{x_m}, \mathbf{x_n}) + (\mathbf{1} - \mathbf{a_{m,n}}) \log(\mathbf{1} - \rho(\mathbf{x_u}, \mathbf{x_v})))$. Here, note that we used the function $\rho(\mathbf{x}_u, \mathbf{x}_v) = \exp(-\frac{1}{2}\|\mathbf{x}_u - \mathbf{x}_v\|^2)$ in our experiments.

4.3 Experimental Results

We first evaluated the efficiency of our proposed method in comparison with the existing methods. We show our experimental results in Fig. 3, where Spec, MDS, SF, CE, eSE and pSE stand for the spectral embedding, multi-dimensional scaling, spring force embedding, cross-entropy embedding, existing spherical embedding and proposed spherical embedding methods, respectively (machine used is Intel(R) Xeon(R) CPU X5472 @3.0GHz with 64GB memory). Here Figs. 3a and 3b correspond to the results by using the bipartite graphs constructed from the Yahoo!Movies and Yahoo!Answers sites, respectively. In these figures, we plotted the average processing time (sec.) over 10 trials by changing the initial position vectors, where the horizontal and vertical axes stand for the number of nodes in V_B and the processing times, respectively. Here recall that the number of nodes in V_B is different for each bipartite graph as mentioned above.

As expected, these figures show that our proposed spherical embedding (pSE) method works much faster than all the existing methods we compared. More specifically, the spectral embedding (Spec) method works comparable to our method. This is because these methods perform a power iteration on a sparse adjacency matrix. In fact, the multi-dimensional scaling (MDS) method requires a substantially large computation time because it needs to perform a power iteration on a full distance matrix. All the other methods including the existing spherical embedding (eSE) method, which repeatedly move each position vector by using the Newton method, generally require an extremely



(a) Yahoo!Movies

(b) Yahoo!Answers

Fig. 3: Comparison of processing times.

large computation time before the final results are obtained. Especially, both the spring force embedding (SF) and cross-entropy embedding (SE) methods require more than three days to obtain the final results even for one trial when the numbers of nodes for the Yahoo!Answers graphs become more than 40,000; thus we omitted these results in Fig. 3b. Here we should emphasize that the scale of the vertical axis of these figures is logarithmic.

Next we evaluated the visualization results of our proposed method in comparison with the existing methods. Due to a space limitation, we only show our experimental results obtained for a bipartite graph constructed from the Japanese Yahoo!Movies sites in Fig. 4. Here recall that the genre information has been shown in Fig. 2. In Fig. 4a, we show the visualization result by our proposed method, which we consider intuitively natural. Actually, we can see that the genre nodes of Action/Adventure (black square) and Suspense (maroon hexagon) are located in near positions at the right-side of the inner circle (θ_A), while at the opposite left-side of this circle, the genre nodes of Teen (orange triangle_up) and Romance (cyan square) are located in near positions. Overall, we can observe that the similar genres are located closely on the inner circle (θ_A).

Now we compare the above results with the five existing methods. The first one is the visualization result by the existing spherical embedding method shown in Fig. 4b. We see that there are several minor differences but we consider this result comparable to the result by our method. However, this one is very slow and inefficient. Our method is much faster. The second one is the visualization result by the multidimensional scaling method shown in Fig. 4c. We can observe some clusters of genres. Although this result might indicate some intrinsic property, we feel that the spherical embedding scheme is a more natural representation of bipartite graphs. The third one is the visualization result by the spectral embedding method shown in Fig. 4d. This one is relatively poor in our own experiments. In fact, the two genres of Drama (lime cross) at the bottom-right and Documentary (Olive triangle_right) at the top-left are too much isolated, although this method works reasonably fast among the existing methods.The fourth and the fifth ones are the visualization results by the spring force embedding method and the cross-entropy embedding method shown in Figs. 4e and 4f. We can observe a similar tendency between these two, *e.g.*, we can easily see that the genre node of Drama (lime cross) is much isolated in both. The main difference in these methods is that we can observe that some genre nodes are clustered for the spring force embedding method, but there are no such clusters and all the genres are scattered for the cross-entropy embedding method. Overall, although each embedding method might have its own characteristics that are both advantageous and disadvantageous, we believe that our proposed spherical embedding method is most effective for visualizing bipartite graphs in terms of efficiency and interpretability.

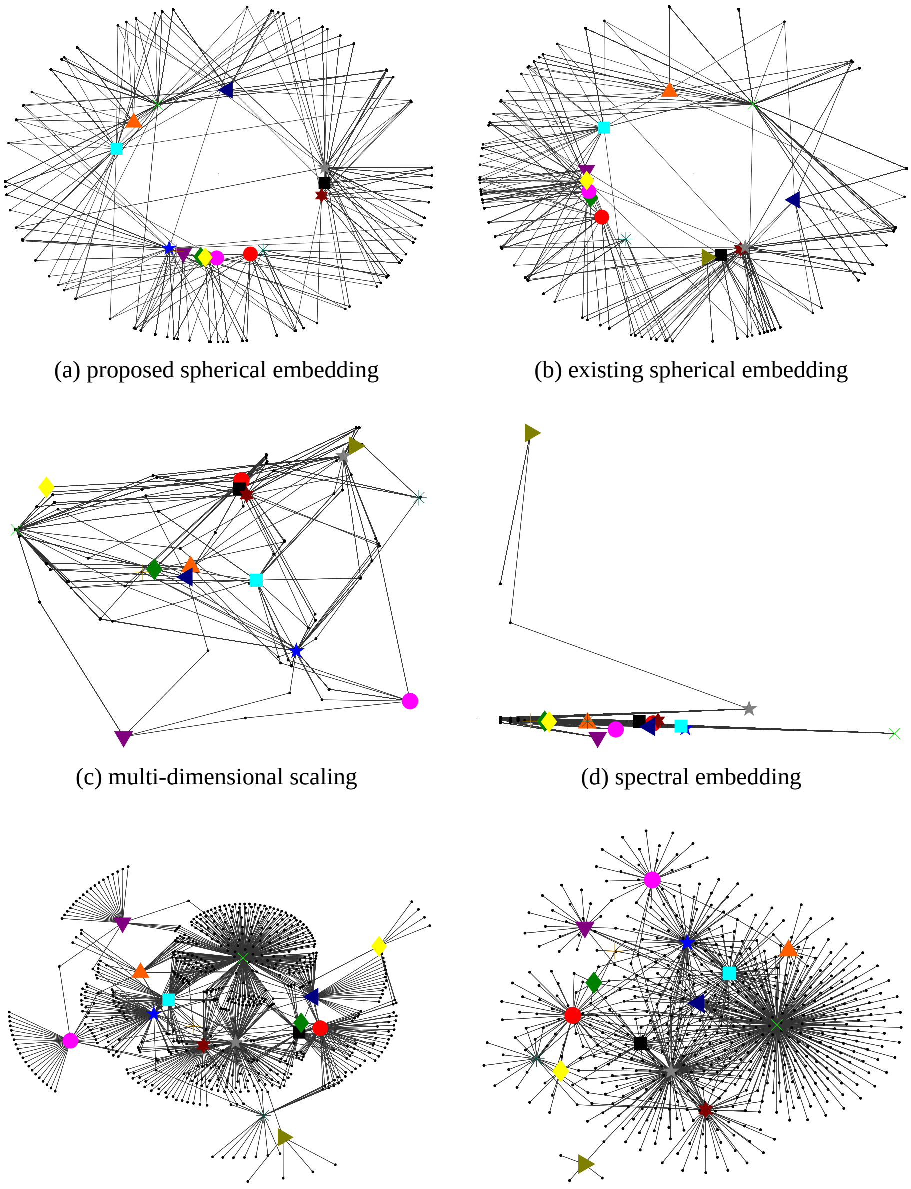
Last but not least, we evaluated our proposed method only in the case of two-dimensional embedding for our visualization purpose, but this does not mean that it is limited to two-dimensional embedding. It is quite easy to extend it to the general K-dimension embedding. We plan to evaluate our method as a powerful technique for both dimensional reduction and clustering as a future work.

5 Conclusion

In this paper, we addressed the problem of visualizing structure of bipartite graphs such as relations between pairs of objects and their multi-labeled categories, and proposed a new spherical embedding method that is based on a power iteration. The key features of this method is that it employs two operations on the positioning vectors, one called double-centering operation and the other called normalizing operation. This enables the iterative approach to be equivalent to maximizing an objective function which is guaranteed to converge. Thus, our algorithm is theoretically guaranteed to converge. We applied our method to a set of bipartite graphs with different sizes and connections which were constructed from the Japanese sites of Yahoo!Movies and Yahoo!Answers, and compared the results with five existing visualization methods. The results showed that the proposed method works much faster than all the five existing methods, and the visualization results are intuitively understandable and comparable to the best available so far known. In future, we plan to apply the new method to evaluate its performance for a wide variety of bipartite graphs.

References

1. Chung, F. R. K. (1997). Spectral Graph Theory", Number 92 in CBMS Regional Conference Series in Mathematics, American Mathematical Society.
2. Kamada, K., & Kawai, S. (1989). An algorithm for drawing general undirected graph. *Information Processing Letters*, *31*, 7–15.
3. Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, **46:5**, 604–632.
4. Langville, A. N., & Meyer, C. D. (2005). Deeper inside PageRank, *Internet Mathematics*, **1:3** 335–380.
5. Torgerson, W. (1958). Theory and methods of scaling. Wiley New York.
6. Naud, A., Usui, S., Ueda, N., & Taniguchi, T. (2007). Visualization of documents and concepts in neuroinformatics with the 3D-SE viewer. *Frontiers in Neuroinformatics*, **1, 7**.
7. Yamada, T., Saito, K., & Ueda, N. (2003). Cross-entropy directed embedding of network data. In *Proceedings of the 20th International Conference on Machine Learning* (pp. 832–839).



(a) proposed spherical embedding

(b) existing spherical embedding

(c) multi-dimensional scaling

(d) spectral embedding

(e) spring force embedding

(f) cross-entropy embedding

Fig. 4: Visualization Results (Yahoo!Movies 1950 - 1959)